PATENT COOPERATION TREATY

From the INTERNATIONAL SEARCHING AUTHORITY

To: YASUO MURAMATSU
MURAMATSO & ASSOCIATES
7700 IRVINE CENTER DRIVE
SUITE 225
IRVINE, CA 92618

PCT

NOTIFICATION OF TRANSMITTAL OF THE INTERNATIONAL SEARCH REPORT OR THE DECLARATION

(PCT Rule 44.1)

Date of Mailing *(day/month/year)*	27 DEC 2000
Applicant's or agent's file reference ADVA214.001P	**FOR FURTHER ACTION** **See paragraphs 1 and 4 below**
International application No. PCT/US00/26189	International filing date *(day/month/year)* 23 SEPTEMBER 2000
Applicant ADVANTEST CORPORATION	

1. [X] The applicant is hereby notified that the international search report has been established and is transmitted herewith.

Filing of amendments and statement under Article 19:
The applicant is entitled, if he so wishes, to amend the claims of the international application (see Rule 46):

When? The time limit for filing such amendments is normally 2 months from the date of transmittal of the international search report; however, for more details, see the notes on the accompanying sheet.

Where? Directly to the International Bureau of WIPO
34, chemin des Colombettes
1211 Geneva 20, Switzerland
Facsimile No.: (41-22) 740.14.35

For more detailed instructions, see the notes on the accompanying sheet.

2. [ ] The applicant is hereby notified that no international search report will be established and that the declaration under Article 17(2)(a) to that effect is transmitted herewith.

3. [ ] **With regard to the protest** against payment of (an) additional fee(s) under Rule 40.2, the applicant is notified that:

[ ] the protest together with the decision thereon has been transmitted to the International Bureau together with the applicant's request to forward the texts of both the protest and the decision thereon to the designated Offices.

[ ] no decision has been made yet on the protest; the applicant will be notified as soon as a decision is made.

4. **Further action(s):** The applicant is reminded of the following:

Shortly after **18 months** from the priority date, the international application will be published by the International Bureau. If the applicant wishes to avoid or postpone publication, a notice of withdrawal of the international application, or of the priority claim, must reach the International Bureau as provided in rules 90 *bis* 1 and 90 *bis* 3, respectively, before the completion of the technical preparations for international publication.

Within **19 months** from the priority date, a demand for international preliminary examination must be filed if the applicant wishes to postpone the entry into the national phase until 30 months from the priority date (in some Offices even later).

Within **20 months** from the priority date, the applicant must perform the prescribed acts for entry into the national phase before all designated Offices which have not been elected in the demand or in a later election within 19 months from the priority date or could not be elected because they are not bound by Chapter II.

Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231	Authorized officer Peggy Harrod JOSEPH DAVID TORRES
Facsimile No. (703) 305-3230	Telephone No. (703) - 308 -7066

(See notes on accompanying sheet)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/26189

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y,P	US 6,061,283 (TAKAHASHI et al) 09 May 2000, note especially the reference to STIL and CAD in figure 3 and col. 3, lines 39-67.	8
Y	US 4,894,829 A (MONIE et al) 16 January 1990, col. 7, lines 37-62; col. 8, lines 13-42; figure 4.	1-7
A	US 4,507,740 A (STAR et al) 26 March 1985, col. 2, lines 1-21.	1-8
Y	US 4,870,397 A (SOTO et al) 26 September 1989, col. 2, lines 57-67; col. 3, lines 1-50.	1-2
Y	US 5,046,034 (STARK et al) 03 September 1991, col. 3, lines 39-53.	1-2
Y	US 5,323,401 (MASTON) 21 June 1994, col. 2, lines 13-29; figure 2.	1-2,5,6
X	US 5,845,234 A (TESTA et al) 01 December 1998, col. 2, lines 15-59, figure 3 and 4.	1-2
Y	US 5,862,149 (CARPENTER et al) 19 January 1999, col. 4, lines 24-67; col. 5, lines 1-22; figure 8.	1-4
A,P	US 6,052,809 (BOWDEN) 18 April 2000, col. 2, lines 15-39; figure 2A, 2B and 2C.	1-8
A	US 6,092,225 (GRUODIS et al) 18 July 2000, col. 2, lines 39-67; col. 3, lines 1-28.	1-2

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/26189

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) : G01R 31/28
US CL : 714/724
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 714/724, 725, 738

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X --- Y	US 5,778,004 A (JENNION et al) 07 July 1998, col. 2, lines 56-67; col. 4, lines 1-32; col. 7, lines 14-67; col 8, lines 14-36; figures 5A and 5B.	1-7 ---- 8
X, P ------ Y, P	US 6,067,652 A (FUSCO et al) 23 May 2000, col. 2, lines 30-64; col. 3, lines 64-67; col. 4, lines 1-45; col. 4, lines 59-67; col. 5, lines 1-45; col. 6, lines 37-67; figures 2A, 2B, 3A, 3B and 4.	1-7 ---- 8
X,P --- Y,P	US 5,974,241 A (FUSCO et al) 26 October 1999, col. 2, lines 57-67; col. 3, lines 35-59; especially, note reference to CAE/CAD in col. 5, lines 25-33.	1-2, 5-7 ----- 8

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
09 NOVEMBER 2000	27 DEC 2000

Name and mailing address of the ISA/US	Authorized officer
Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231	JOSEPH DAVID TORRES
Facsimile No. (703) 305-3230	Telephone No. (703) - 308 -7066

PATENT COOPERATION TREATY

PCT

INTERNATIONAL SEARCH REPORT

(PCT Article 18 and Rules 43 and 44)

Applicant's or agent's file reference ADVA214.001P	FOR FURTHER ACTION	see Notification of Transmittal of International Search Report (Form PCT/ISA/220) as well as, where applicable, item 5 below.
International application No. PCT/US00/26189	International filing date *(day/month/year)* 23 SEPTEMBER 2000	(Earliest) Priority Date *(day/month/year)* 25 SEPTEMBER 1999
Applicant ADVANTEST CORPORATION		

This international search report has been prepared by this International Searching Authority and is transmitted to the applicant according to Article 18. A copy is being transmitted to the International Bureau.

This international search report consists of a total of 4 sheets.

[X] It is also accompanied by a copy of each prior art document cited in this report.

1. **Basis of the report**

   a. With regard to the **language**, the international search was carried out on the basis of the international application in the language in which it was filed, unless otherwise indicated under this item.

   [ ] the international search was carried out on the basis of a translation of the international application furnished to this Authority (Rule 23.1(b)).

   b. With regard to any **nucleotide and/or amino acid sequence** disclosed in the international application, the international search was carried out on the basis of the sequence listing:

   [ ] contained in the international application in written form.

   [ ] filed together with the international application in computer readable form.

   [ ] furnished subsequently to this Authority in written form.

   [ ] furnished subsequently to this Authority in computer readable form.

   [ ] the statement that the subsequently furnished written sequence listing does not go beyond the disclosure in the international application as filed has been furnished.

   [ ] the statement that the information recorded in computer readable form is identical to the written sequence listing has been furnished.

2. [ ] **Certain claims were found unsearchable** (See Box I).

3. [ ] **Unity of invention is lacking** (See Box II).

4. With regard to the **title**,

   [X] the text is approved as submitted by the applicant.

   [ ] the text has been established by this Authority to read as follows:

5. With regard to the **abstract**,

   [ ] the text is approved as submitted by the applicant.

   [X] the text has been established, according to Rule 38.2(b), by this Authority as it appears in Box III. The applicant may, within one month from the date of mailing of this international search report, submit comments to this Authority.

6. The figure of the **drawings** to be published with the abstract is Figure No. 6B

   [ ] as suggested by the applicant.

   [X] because the applicant failed to suggest a figure.

   [ ] because this figure better characterizes the invention.

   [ ] None of the figures.

Box III TEXT OF THE ABSTRACT (Continuation of item 5 of the first sheet)

A method of converting test vectors (21) in an original test-cycle based language into a target-cycle based test language (22). The method includes the steps of forming a set of templates (25) depicting waveforms defined in the target test language (22); decomposing a waveform (24) in the original test language into a set of constituent events where each event includes data showing at least a starting value and a number of subsequent edges of the waveform; comparing the template and the set of events (26); storing the waveform data in the target test language when a match is detected (28) and retrieving corresponding parameters of the waveform in the original test language (29); and repeating the above steps for all of the test vectors in the original test language, therby forming a test vector file of the target test language.

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/26189

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) : G01R 31/28
US CL : 714/724
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 714/724, 725, 738

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X --- Y	US 5,778,004 A (JENNION et al) 07 July 1998, col. 2, lines 56-67; col. 4, lines 1-32; col. 7, lines 14-67; col 8, lines 14-36; figures 5A and 5B.	1-7 ---- 8
X, P ------ Y, P	US 6,067,652 A (FUSCO et al) 23 May 2000, col. 2, lines 30-64; col. 3, lines 64-67; col. 4, lines 1-45; col. 4, lines 59-67; col. 5, lines 1-45; col. 6, lines 37-67; figures 2A, 2B, 3A, 3B and 4.	1-7 ---- 8
X,P --- Y,P	US 5,974,241 A (FUSCO et al) 26 October 1999, col. 2, lines 57-67; col. 3, lines 35-59; especially, note reference to CAE/CAD in col. 5, lines 25-33.	1-2, 5-7 ----- 8

[X] Further documents are listed in the continuation of Box C.   [ ] See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

09 NOVEMBER 2000

Date of mailing of the international search report

27 DEC 2000

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231
Facsimile No. (703) 305-3230

Authorized officer
Peggy Harrod
JOSEPH DAVID TORRES
Telephone No. (703) 308-7066

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y,P	US 6,061,283 (TAKAHASHI et al) 09 May 2000, note especially the reference to STIL and CAD in figure 3 and col. 3, lines 39-67.	8
Y	US 4,894,829 A (MONIE et al) 16 January 1990, col. 7, lines 37-62; col. 8, lines 13-42; figure 4.	1-7
A	US 4,507,740 A (STAR et al) 26 March 1985, col. 2, lines 1-21.	1-8
Y	US 4,870,397 A (SOTO et al) 26 September 1989, col. 2, lines 57-67; col. 3, lines 1-50.	1-2
Y	US 5,046,034 (STARK et al) 03 September 1991, col. 3, lines 39-53.	1-2
Y	US 5,323,401 (MASTON) 21 June 1994, col. 2, lines 13-29; figure 2.	1-2,5,6
X	US 5,845,234 A (TESTA et al) 01 December 1998, col. 2, lines 15-59, figure 3 and 4.	1-2
Y	US 5,862,149 (CARPENTER et al) 19 January 1999, col. 4, lines 24-67; col. 5, lines 1-22; figure 8.	1-4
A,P	US 6,052,809 (BOWDEN) 18 April 2000, col. 2, lines 15-39; figure 2A, 2B and 2C.	1-8
A	US 6,092,225 (GRUODIS et al) 18 July 2000, col. 2, lines 39-67; col. 3, lines 1-28.	1-2

THIS PAGE BLANK (USPTO)

3

PATENT COOPERATION TREATY

PCT

INTERNATIONAL PRELIMINARY EXAMINATION REPORT

(PCT Article 36 and Rule 70)

Applicant's or agent's file reference ADVA214.001P	**FOR FURTHER ACTION** See Notification of Transmittal of International Preliminary Examination Report (Form PCT/IPEA/416)	
International application No. PCT/US00/26189	International filing date *(day/month/year)* 23 September 2000 (23.09.2000)	Priority date *(day/month/year)* 25 September 1999 (25.09.1999)
International Patent Classification (IPC) or national classification and IPC IPC(7): G01R 31/28 and US Cl.: 714/724		
Applicant ADVANTEST CORPORATION		

1. This international preliminary examination report has been prepared by this International Preliminary Examining Authority and is transmitted to the applicant according to Article 36.

2. This REPORT consists of a total of 6 sheets, including this cover sheet.

   ☐ This report is also accompanied by ANNEXES, i.e., sheets of the description, claims and/or drawings which have been amended and are the basis for this report and/or sheets containing rectifications made before this Authority (see Rule 70.16 and Section 607 of the Administrative Instructions under the PCT).

   These annexes consist of a total of Ø sheets.

3. This report contains indications relating to the following items:

   - I ☒ Basis of the report
   - II ☐ Priority
   - III ☐ Non-establishment of report with regard to novelty, inventive step and industrial applicability
   - IV ☐ Lack of unity of invention
   - V ☒ Reasoned statement under Article 35(2) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement
   - VI ☐ Certain documents cited
   - VII ☐ Certain defects in the international application
   - VIII ☐ Certain observations on the international application

Date of submission of the demand 21 April 2001 (21.04.2001)	Date of completion of this report 24 September 2001 (24.09.2001)
Name and mailing address of the IPEA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703)305-3230	Authorized officer Albert Decady Peggy Harrod Telephone No. (703)305-3900

INTERNATIONAL PRELIMINARY EXAMINATION REPORT

International application No.
PCT/US00/26189

I. Basis of the report

1. With regard to the elements of the international application:*

☒ the international application as originally filed.

☒ the description:
pages 1-23 as originally filed
pages NONE, filed with the demand
pages NONE, filed with the letter of _____.

☒ the claims:
pages 24-26, as originally filed
pages NONE, as amended (together with any statement) under Article 19
pages NONE, filed with the demand
pages NONE, filed with the letter of _____.

☒ the drawings:
pages 1-15, as originally filed
pages NONE, filed with the demand
pages NONE, filed with the letter of _____.

☒ the sequence listing part of the description:
pages NONE, as originally filed
pages NONE, filed with the demand
pages NONE, filed with the letter of _____.

2. With regard to the **language**, all the elements marked above were available or furnished to this Authority in the language in which the international application was filed, unless otherwise indicated under this item.
These elements were available or furnished to this Authority in the following language _____ which is:

☐ the language of a translation furnished for the purposes of international search (under Rule23.1(b)).

☐ the language of publication of the international application (under Rule 48.3(b)).

☐ the language of the translation furnished for the purposes of international preliminary examination(under Rules 55.2 and/or 55.3).

3. With regard to any **nucleotide and/or amino acid sequence** disclosed in the international application, the international preliminary examination was carried out on the basis of the sequence listing:

☐ contained in the international application in printed form.

☐ filed together with the international application in computer readable form.

☐ furnished subsequently to this Authority in written form.

☐ furnished subsequently to this Authority in computer readable form.

☐ The statement that the subsequently furnished written sequence listing does not go beyond the disclosure in the international application as filed has been furnished.

☐ The statement that the information recorded in computer readable form is identical to the written sequence listing has been furnished.

4. ☒ The amendments have resulted in the cancellation of:

☒ the description, pages NONE

☒ the claims, Nos. NONE

☒ the drawings, sheets/~~fig~~ NONE

5. ☐ This report has been established as if (some of) the amendments had not been made, since they have been considered to go beyond the disclosure as filed, as indicated in the Supplemental Box (Rule 70.2(c)).**

* *Replacement sheets which have been furnished to the receiving Office in response to an invitation under Article 14 are referred to in this report as "originally filed" and are not annexed to this report since they do not contain amendments (Rules 70.16 and 70.17).*
** *Any replacement sheet containing such amendments must be referred to under item 1 and annexed to this report.*

INTERNATIONAL PRELIMINARY EXAMINATION REPORT

International application No.
PCT/US00/26189

V. Reasoned statement under Rule 66.2(a)(ii) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement

1. STATEMENT

Novelty (N)	Claims	3-8	YES
	Claims	1,2	NO
Inventive Step (IS)	Claims	NONE	YES
	Claims	1-8	NO
Industrial Applicability (IA)	Claims	1-8	YES
	Claims	NONE	NO

2. CITATIONS AND EXPLANATIONS

Please See Continuation Sheet

INTERNATIONAL PRELIMINARY EXAMINATION REPORT

International application No.
PCT/US00/26189

Supplemental Box
(To be used when the space in any of the preceding boxes is not sufficient)

Continuation of Certain Documents Cited

1. Certain published documents (Rule 70.10)

Application No Patent No.	Publication Date (day/month/year)	Filing Date (day/month/year)	Priority date (valid claim) (day/month/year)
None	None	None	None

2. Non-written disclosures (Rule 70.9)

Kind of non-written disclosure	Date of non-written disclosure (day/month/year)	Date of written disclosure referring to non-written disclosure (day/month/year)
None	None	None

Supplemental Box
(To be used when the space in any of the preceding boxes is not sufficient)

Claims 1 and 2 lack novelty under PCT Article 33(2) as being anticipated by Testa et al. (US 5845234, hereafter referred to as Testa).

Testa teaches "The cyclized waveform data 22 is converted into an ATE testing program code 24 containing format and timing information for use in a wide variety of ATE implementations" (col. 4, lines 44-56, Testa).

Testa teaches "FIG. 2 is a block diagram of a prior art method for generating the testing program code 12 for use in the automatic test equipment 11. Cyclized waveform data 22 is provided as program inputs. Generally, the cyclized waveform data 22 consist of waveform-based pattern data associated with waveform templates. Each set of pattern data is defined for a number of events or signals. Each waveform template defines a reusable time slice which can be associated with multiple data pattern rows. Time slices are commonly referred to as "cycles." In essence, each waveform template describes what a set of electrical signals do at specific times relative to the start of a cycle. Specifications are designated for each waveform template and are usually defined for, but not limited to, a series of edge times associated with specific events. Examples of events include driving a specific logic level, turning off a driver, sampling data, driving a logic level specified by the pattern data associated with the cycle or other related functions" (col. 4, lines 19-35, Testa).

Testa teaches "FIG. 4 shows a data structure for storing the cyclized waveform data 22 used in the method of FIG. 3. In the described embodiment, the cyclized waveform data 22 is expressed in a waveform description language as is known in the art. The cyclized waveform data 22 stores cyclized data 40, pattern information 41 and ATE channel and device-under-test (DUT) pin information 42 as described in the following sections (col. 5, lines 29-36, Testa).Testa teaches "The cyclized data 40 is stored according to a standard syntax for describing the waveform event data and pattern information. The standard syntax is based on a modified STIL character set as shown below in Table 1" which include: E Compare Edge Strobe to Pattern Value, W Compare Window Strobe to Pattern Value, R Compare Strobe to Inverse Pattern Value, T Compare Tristate (col. 5, lines 38-58, Testa).

Claim 3 lacks an inventive step under PCT Article 33(3) as being obvious over Testa (US 5845234). Testa substantially teaches the claimed invention described in claims 1 and 2 (see previous comments above). However Testa does not teach the step of storing test data in a table consisting of start time and subsequent edges. Testa teaches "each waveform template describes what a set of electrical signals do at specific times relative to the start of a cycle" and "waveform template and are usually defined for, but not limited to, a series of edge times associated with specific events" (col. 4, lines 19-35, Testa). Testa teaches "FIG. 4 shows a data structure for storing the cyclized waveform" (col. 5, lines 29-36, Testa). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Testa by including an additional step of storing test data in a table consisting of start time and subsequent edges. This modification would have been obvious to one of ordinary skill in the art, at the time of the invention, because one of ordinary skill in the art at the time of the invention would have recognized that storing test data in a table consisting of start time and subsequent edges would enhance testing.

Claims 4-8 lacks an inventive step under PCT Article 33(3) as being obvious over Testa (US 5845234).
Testa teaches the additional limitations of claim 4. See comments, in writen opinion, to claim 3, above.

Testa teaches the additional limitations of claim 5. See comments, in writen opinion, to claim 3, above. Testa teaches "The

Supplemental Box
(To be used when the space in any of the preceding boxes is not sufficient)

cyclized data 40 is stored according to a standard syntax for describing the waveform event data and pattern information. The standard syntax is based on a modified STIL character set as shown below in Table 1" which include: E Compare Edge Strobe to Pattern Value, W Compare Window Strobe to Pattern Value, R Compare Strobe to Inverse Pattern Value, T Compare Tristate (col. 5, lines 38-58, Testa). Testa teaches "data is in a form that allows simple mapping of the information into list driven templates for generating the output ATE test program modules (col. 11, lines 51-65, Testa).

Testa teaches the additional limitations of claim 6. Testa teaches "each waveform template describes what a set of electrical signals do at specific times relative to the start of a cycle" and "waveform template and are usually defined for, but not limited to, a series of edge times associated with specific events" (col. 4, lines 19-35, Testa).

Testa teaches the additional limitations of claim 7. Testa teaches " the signals can be grouped into higher level constructs which include: (1) buses 49A-C, and (2) multiplexed groups 50A-C. Each bus 49A is an indexed list of signals all associated with one name. Each multiplexed group 50A is a grouping of two or more signals which are grouped together to show combined signal information (col. 7, lines 5-10, Testa).

Testa teaches the additional limitations of claim 8. STIL is a cycle based test language with all the properties of the cycle based wavforms in Testa. Converting STIL waveforms to ATE testing code is encompassed by Testa's invention since Testa teaches converting cycle based waveforms to ATE testing code and as such converting STIL waveforms to ATE testing code does not depart from the intended scope of Testa's invention.

Claims 1-8 meet industrial applicability as defined by PCT Article 33(4).

---------------- NEW CITATIONS ----------------

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/26189

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) : G01R 31/28
US CL : 714/724
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 714/724, 725, 738

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X --- Y	US 5,778,004 A (JENNION et al) 07 July 1998, col. 2, lines 56-67; col. 4, lines 1-32; col. 7, lines 14-67; col 8, lines 14-36; figures 5A and 5B.	1-7 ---- 8
X, P ------ Y, P	US 6,067,652 A (FUSCO et al) 23 May 2000, col. 2, lines 30-64; col. 3, lines 64-67; col. 4, lines 1-45; col. 4, lines 59-67; col. 5, lines 1-45; col. 6, lines 37-67; figures 2A, 2B, 3A, 3B and 4.	1-7 ---- 8
X,P --- Y,P	US 5,974,241 A (FUSCO et al) 26 October 1999, col. 2, lines 57-67; col. 3, lines 35-59; especially, note reference to CAE/CAD in col. 5, lines 25-33.	1-2, 5-7 ----- 8

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
09 NOVEMBER 2000	27 DEC 2000
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer Peggy Harrod JOSEPH DAVID TORRES Telephone No. (703) - 308 -7066

Form PCT/ISA/210 (second sheet) (July 1998)★

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y,P	US 6,061,283 (TAKAHASHI et al) 09 May 2000, note especially the reference to STIL and CAD in figure 3 and col. 3, lines 39-67.	8
Y	US 4,894,829 A (MONIE et al) 16 January 1990, col. 7, lines 37-62; col. 8, lines 13-42; figure 4.	1-7
A	US 4,507,740 A (STAR et al) 26 March 1985, col. 2, lines 1-21.	1-8
Y	US 4,870,397 A (SOTO et al) 26 September 1989, col. 2, lines 57-67; col. 3, lines 1-50.	1-2
Y	US 5,046,034 (STARK et al) 03 September 1991, col. 3, lines 39-53.	1-2
Y	US 5,323,401 (MASTON) 21 June 1994, col. 2, lines 13-29; figure 2.	1-2,5,6
X	US 5,845,234 A (TESTA et al) 01 December 1998, col. 2, lines 15-59, figure 3 and 4.	1-2
Y	US 5,862,149 (CARPENTER et al) 19 January 1999, col. 4, lines 24-67; col. 5, lines 1-22; figure 8.	1-4
A,P	US 6,052,809 (BOWDEN) 18 April 2000, col. 2, lines 15-39; figure 2A, 2B and 2C.	1-8
A	US 6,092,225 (GRUODIS et al) 18 July 2000, col. 2, lines 39-67; col. 3, lines 1-28.	1-2

M

PATENT COOPERATION TREATY

From the
INTERNATIONAL PRELIMINA[illegible]XAMINING AUTHORITY

To:
YASUO MURAMATSU
MURAMATSU & ASSOCIATES
7700 IRVINE CENTER DRIVE
SUITE 225
IRVINE, CA 92618

PCT

NOTIFICATION OF TRANSMITTAL OF INTERNATIONAL PRELIMINARY EXAMINATION REPORT

(PCT Rule 71.1)

Date of Mailing (*day/month/year*) 29 OCT 2001

Applicant's or agent's file reference
ADVA214.001P

IMPORTANT NOTIFICATION

International application No.	International filing date (*day/month/year*)	Priority date (*day/month/year*)
PCT/US00/26189	23 September 2000 (23.09.2000)	25 September 1999 (25.09.1999)

Applicant
ADVANTEST CORPORATION

1. The applicant is hereby notified that this International Preliminary Examining Authority transmits herewith the international preliminary examination report and its annexes, if any, established on the international application.

2. A copy of the report and its annexes, if any, is being transmitted to the International Bureau for communication to all the elected Offices.

3. Where required by any of the elected Offices, the International Bureau will prepare an English translation of the report (but not of any annexes) and will transmit such translation to those Offices.

4. **REMINDER**

The applicant must enter the national phase before each elected Office by performing certain acts (filing translations and paying national fees) within 30 months from the priority date (or later in some Offices)(Article 39(1))(see also the reminder sent by the International Bureau with Form PCT/IB/301).

Where a translation of the international application must be furnished to an elected Office, that translation must contain a translation of any annexes to the international preliminary examination report. It is the applicant's responsibility to prepare and furnish such translation directly to each elected Office concerned.

For further details on the applicable time limits and requirements of the elected Offices, see Volume II of the PCT Applicant's Guide.

Name and mailing address of the IPEA/US	Authorized officer
Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231	Albert Decady Peggy Harrod
Facsimile No. (703)305-3230	Telephone No. (703)305-3900

PATENT COOPERATION TREATY

PCT

INTERNATIONAL PRELIMINARY EXAMINATION REPORT

(PCT Article 36 and Rule 70)

Applicant's or agent's file reference ADVA214.001P	**FOR FURTHER ACTION**	See Notification of Transmittal of International Preliminary Examination Report (Form PCT/IPEA/416)
International application No. PCT/US00/26189	International filing date *(day/month/year)* 23 September 2000 (23.09.2000)	Priority date *(day/month/year)* 25 September 1999 (25.09.1999)
International Patent Classification (IPC) or national classification and IPC IPC(7): G01R 31/28 and US Cl.: 714/724		
Applicant ADVANTEST CORPORATION		

1. This international preliminary examination report has been prepared by this International Preliminary Examining Authority and is transmitted to the applicant according to Article 36.

2. This REPORT consists of a total of 6 sheets, including this cover sheet.

   ☐ This report is also accompanied by ANNEXES, i.e., sheets of the description, claims and/or drawings which have been amended and are the basis for this report and/or sheets containing rectifications made before this Authority (see Rule 70.16 and Section 607 of the Administrative Instructions under the PCT).

   These annexes consist of a total of ø sheets.

3. This report contains indications relating to the following items:

   I ☒ Basis of the report
   II ☐ Priority
   III ☐ Non-establishment of report with regard to novelty, inventive step and industrial applicability
   IV ☐ Lack of unity of invention
   V ☒ Reasoned statement under Article 35(2) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement
   VI ☐ Certain documents cited
   VII ☐ Certain defects in the international application
   VIII ☐ Certain observations on the international application

Date of submission of the demand 21 April 2001 (21.04.2001)	Date of completion of this report 24 September 2001 (24.09.2001)
Name and mailing address of the IPEA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703)305-3230	Authorized officer Peggy Harrod Albert Decady Telephone No. (703)305-3900

Form PCT/IPEA/409 (cover sheet)(July 1998)

THIS PAGE BLANK (USPTO)

INTERNATIONAL PRELIMINARY EXAMINATION REPORT

International application No.
PCT/US00/26189

I. Basis of the report

1. With regard to the elements of the international application:*

☒ the international application as originally filed.

☒ the description:
pages 1-23 as originally filed
pages NONE, filed with the demand
pages NONE, filed with the letter of ________.

☒ the claims:
pages 24-26, as originally filed
pages NONE, as amended (together with any statement) under Article 19
pages NONE, filed with the demand
pages NONE, filed with the letter of ________.

☒ the drawings:
pages 1-15, as originally filed
pages NONE, filed with the demand
pages NONE, filed with the letter of ________.

☒ the sequence listing part of the description:
pages NONE, as originally filed
pages NONE, filed with the demand
pages NONE, filed with the letter of ________.

2. With regard to the **language**, all the elements marked above were available or furnished to this Authority in the language in which the international application was filed, unless otherwise indicated under this item.
These elements were available or furnished to this Authority in the following language _____ which is:

☐ the language of a translation furnished for the purposes of international search (under Rule23.1(b)).

☐ the language of publication of the international application (under Rule 48.3(b)).

☐ the language of the translation furnished for the purposes of international preliminary examination(under Rules 55.2 and/or 55.3).

3. With regard to any **nucleotide and/or amino acid sequence** disclosed in the international application, the international preliminary examination was carried out on the basis of the sequence listing:

☐ contained in the international application in printed form.

☐ filed together with the international application in computer readable form.

☐ furnished subsequently to this Authority in written form.

☐ furnished subsequently to this Authority in computer readable form.

☐ The statement that the subsequently furnished written sequence listing does not go beyond the disclosure in the international application as filed has been furnished.

☐ The statement that the information recorded in computer readable form is identical to the written sequence listing has been furnished.

4. ☒ The amendments have resulted in the cancellation of:

☒ the description, pages NONE

☒ the claims, Nos. NONE

☒ the drawings, sheets/~~fig~~ NONE

5. ☐ This report has been established as if (some of) the amendments had not been made, since they have been considered to go beyond the disclosure as filed, as indicated in the Supplemental Box (Rule 70.2(c)).**

* *Replacement sheets which have been furnished to the receiving Office in response to an invitation under Article 14 are referred to in this report as "originally filed" and are not annexed to this report since they do not contain amendments (Rules 70.16 and 70.17).*
** *Any replacement sheet containing such amendments must be referred to under item 1 and annexed to this report.*

INTERNATIONAL PRELIMINARY EXAMINATION REPORT

International application No.
PCT/US00/26189

V. Reasoned statement under Rule 66.2(a)(ii) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement

1. STATEMENT

Novelty (N)	Claims 3-8	YES
	Claims 1,2	NO
Inventive Step (IS)	Claims NONE	YES
	Claims 1-8	NO
Industrial Applicability (IA)	Claims 1-8	YES
	Claims NONE	NO

2. CITATIONS AND EXPLANATIONS

Please See Continuation Sheet

INTERNATIONAL PRELIMINARY EXAMINATION REPORT

International application No.
PCT/US00/26189

Supplemental Box
(To be used when the space in any of the preceding boxes is not sufficient)

Continuation of Certain Documents Cited

1. Certain published documents (Rule 70.10)

Application No Patent No.	Publication Date (day/month/year)	Filing Date (day/month/year)	Priority date (valid claim) (day/month/year)
None	None	None	None

2. Non-written disclosures (Rule 70.9)

Kind of non-written disclosure	Date of non-written disclosure (day/month/year)	Date of written disclosure referring to non-written disclosure (day/month/year)
None	None	None

:HIS PAGE BLANK (USPTO)

INTERNATIONAL PRELIMINARY EXAMINATION REPORT

International application No.
PCT/US00/26189

Supplemental Box
(To be used when the space in any of the preceding boxes is not sufficient)

Claims 1 and 2 lack novelty under PCT Article 33(2) as being anticipated by Testa et al. (US 5845234, hereafter referred to as Testa).

Testa teaches "The cyclized waveform data 22 is converted into an ATE testing program code 24 containing format and timing information for use in a wide variety of ATE implementations" (col. 4, lines 44-56, Testa).

Testa teaches "FIG. 2 is a block diagram of a prior art method for generating the testing program code 12 for use in the automatic test equipment 11. Cyclized waveform data 22 is provided as program inputs. Generally, the cyclized waveform data 22 consist of waveform-based pattern data associated with waveform templates. Each set of pattern data is defined for a number of events or signals. Each waveform template defines a reusable time slice which can be associated with multiple data pattern rows. Time slices are commonly referred to as "cycles." In essence, each waveform template describes what a set of electrical signals do at specific times relative to the start of a cycle. Specifications are designated for each waveform template and are usually defined for, but not limited to, a series of edge times associated with specific events. Examples of events include driving a specific logic level, turning off a driver, sampling data, driving a logic level specified by the pattern data associated with the cycle or other related functions" (col. 4, lines 19-35, Testa).

Testa teaches "FIG. 4 shows a data structure for storing the cyclized waveform data 22 used in the method of FIG. 3. In the described embodiment, the cyclized waveform data 22 is expressed in a waveform description language as is known in the art. The cyclized waveform data 22 stores cyclized data 40, pattern information 41 and ATE channel and device-under-test (DUT) pin information 42 as described in the following sections (col. 5, lines 29-36, Testa).Testa teaches "The cyclized data 40 is stored according to a standard syntax for describing the waveform event data and pattern information. The standard syntax is based on a modified STIL character set as shown below in Table 1" which include: E Compare Edge Strobe to Pattern Value, W Compare Window Strobe to Pattern Value, R Compare Strobe to Inverse Pattern Value, T Compare Tristate (col. 5, lines 38-58, Testa).

Claim 3 lacks an inventive step under PCT Article 33(3) as being obvious over Testa (US 5845234). Testa substantially teaches the claimed invention described in claims 1 and 2 (see previous comments above). However Testa does not teach the step of storing test data in a table consisting of start time and subsequent edges. Testa teaches "each waveform template describes what a set of electrical signals do at specific times relative to the start of a cycle" and "waveform template and are usually defined for, but not limited to, a series of edge times associated with specific events" (col. 4, lines 19-35, Testa). Testa teaches "FIG. 4 shows a data structure for storing the cyclized waveform" (col. 5, lines 29-36, Testa). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Testa by including an additional step of storing test data in a table consisting of start time and subsequent edges. This modification would have been obvious to one of ordinary skill in the art, at the time of the invention, because one of ordinary skill in the art at the time of the invention would have recognized that storing test data in a table consisting of start time and subsequent edges would enhance testing.

Claims 4-8 lacks an inventive step under PCT Article 33(3) as being obvious over Testa (US 5845234).
Testa teaches the additional limitations of claim 4. See comments, in writen opinion, to claim 3, above.

Testa teaches the additional limitations of claim 5. See comments, in writen opinion, to claim 3, above. Testa teaches "The

Supplemental Box
(To be used when the space in any of the preceding boxes is not sufficient)

cyclized data 40 is stored according to a standard syntax for describing the waveform event data and pattern information. The standard syntax is based on a modified STIL character set as shown below in Table 1" which include: E Compare Edge Strobe to Pattern Value, W Compare Window Strobe to Pattern Value, R Compare Strobe to Inverse Pattern Value, T Compare Tristate (col. 5, lines 38-58, Testa). Testa teaches "data is in a form that allows simple mapping of the information into list driven templates for generating the output ATE test program modules (col. 11, lines 51-65, Testa).

Testa teaches the additional limitations of claim 6. Testa teaches "each waveform template describes what a set of electrical signals do at specific times relative to the start of a cycle" and "waveform template and are usually defined for, but not limited to, a series of edge times associated with specific events" (col. 4, lines 19-35, Testa).

Testa teaches the additional limitations of claim 7. Testa teaches " the signals can be grouped into higher level constructs which include: (1) buses 49A-C; and (2) multiplexed groups 50A-C. Each bus 49A is an indexed list of signals all associated with one name. Each multiplexed group 50A is a grouping of two or more signals which are grouped together to show combined signal information (col. 7, lines 5-10, Testa).

Testa teaches the additional limitations of claim 8. STIL is a cycle based test language with all the properties of the cycle based wavforms in Testa. Converting STIL waveforms to ATE testing code is encompassed by Testa's invention since Testa teaches converting cycle based waveforms to ATE testing code and as such converting STIL waveforms to ATE testing code does not depart from the intended scope of Testa's invention.

Claims 1-8 meet industrial applicability as defined by PCT Article 33(4).

------------------- NEW CITATIONS -------------------

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
5 April 2001 (05.04.2001)

PCT

(10) International Publication Number
WO 01/23900 A1

(51) International Patent Classification[7]: G01R 31/28

(21) International Application Number: PCT/US00/26189

(22) International Filing Date: 23 September 2000 (23.09.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/156,121 25 September 1999 (25.09.1999) US

(71) Applicants *(for all designated States except US)*: **ADVANTEST CORPORATION** [JP/JP]; 32-1, Asahi-cho 1-chome, Nerima-ku, Tokyo 179-0071 (JP). **ADVANTEST AMERICA R & D CENTER, INC.** [US/US]; 3201 Scott Boulevard, Santa Clara, CA 95054 (US).

(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: **PARNAS, Bruce, R.** [US/US]; -- (US).

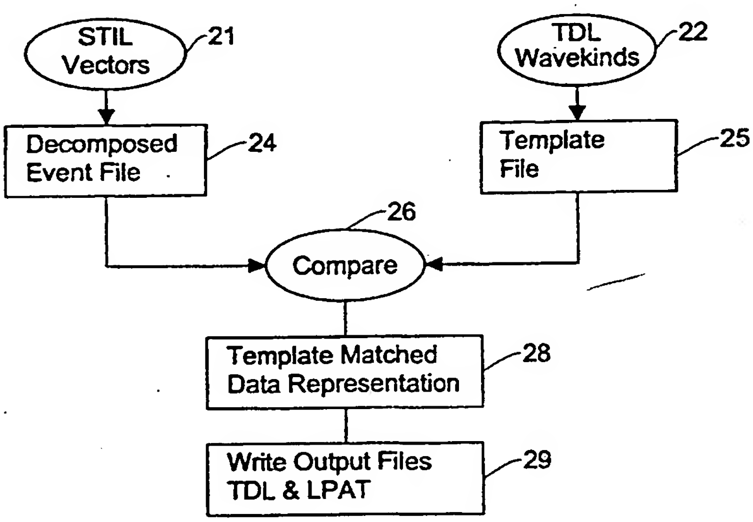
(74) Agent: **MURAMATSU, Yasuo**; Muramatsu & Associates, Suite 225, 7700 Irvine Center Drive, Irvine, CA 92618 (US).

(81) Designated States *(national)*: DE, JP, KR, US.

Published:
— *With international search report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: TEST LANGUAGE CONVERSION METHOD



(57) Abstract: A method of converting test vectors (21) in an original test-cycle based language into a target-cycle based test language (22). The method includes the steps of forming a set of templates (25) depicting waveforms defined in the target test language (22); decomposing a waveform (24) in the original test language into a set of constituent events where each event includes data showing at least a starting value and a number of subsequent edges of the waveform; comparing the template and the set of events (26); storing the waveform data in the target test language when a match is detected (28) and retrieving corresponding parameters of the waveform in the original test language (29); and repeating the above steps for all of the test vectors in the original test language, thereby forming a test vector file of the target test language.

WO 01/23900 A1

TEST LANGUAGE CONVERSION METHOD

Field of the Invention

This invention relates to a test data conversion method for testing semiconductor devices by automatic test equipment, and more particularly, to a method of converting digital test vectors written in STIL (Standard Test Interface Language) to digital test vectors of a test language unique to the automatic test equipment.

Background of the Invention

In testing semiconductor devices such as ICs and LSIs by automatic test equipment (ATE) or an IC tester, a semiconductor IC device to be tested is provided with test signals or test patterns produced by an IC tester at its appropriate pins at predetermined test timings. The IC tester receives output signals from the IC device under test in response to the test signals. The output signals are strobed or sampled by strobe signals generated by the IC tester with predetermined timings to be compared with expected data to determine whether the IC device functions correctly or not. Typically, timings of the test signals and strobe signals are defined relative to a start timing of each test cycle of IC tester.

As noted above, an IC tester generates test patterns and strobes, i.e., test vector, based on digital test vector data described in a language (format) unique to the tester. Such languages for automatic test equipment are different from manufacturer to manufacturer.

Recently, IEEE has proposed a test language STIL (Standard Test Interface Language) as a standard test interface language (IEEE Std 1450-1999). STIL provides an interface between computer-aided engineering (CAE) such as a logic test simulator and automatic test equipment. In a CAE environment or an EDA (electronic design automation) environment, a semiconductor device is designed with the aid

of computer system and such design is tested through a logic test simulator or a testbench. It is preferable to utilize the digital test vectors resulted from the logic simulation in testing actual semiconductor devices by IC testers. STIL is designed to facilitate the transfer of large volumes of digital test vectors from CAE environment to the automatic test equipment (ATE) environment.

The STIL test language has recently become a standard. At present, however, most ATE systems do not use STIL as a native language. Such native languages provided by test equipment manufacturers are not compatible to one another. Accordingly, there is a need to effectively convert STIL to the ATE native language.

Summary of the Invention

It is, therefore, an object of the present invention to provide a method of converting digital test vectors of one cycle based test language to another cycle based test language with high efficiency and accuracy.

It is another object of the present invention to provide a test language conversion method of converting test vectors in a STIL (Standard Test Interface Language) format into a target with high efficiency and accuracy.

In the present invention, the method of converting test vectors in an original cycle based test language into a target cycle based test language, comprising the following steps of: reading available waveforms defined in the target test language and forming a set of templates depicting the waveforms where each template corresponds to a waveform of the target test language and includes data showing at least a starting value of a segment of waveform and a number of subsequent edges in the waveform; reading the test vectors of the original test language and decomposing a waveform in the test vectors in the original test language into a set of constituent events where each event includes data showing at least a starting value and a number of subsequent edges of

the waveform; comparing the template derived from the waveform in the target test language and the set of events derived from the original test language; storing the waveform data in the target test language when a match is detected in the comparison step and retrieving corresponding parameters of the waveform in the test vectors of original test language and storing the parameters in combination with the matched waveform data; and repeating the above steps for all of the test vectors in the original test language, thereby forming a test vector file of the target test language.

Typically, the original test language is STIL (Standard Test Interface Language) specified by IEEE Std 1450-1999. Preferably, the step of comparing the template and the set of events includes a step of changing different levels of test vectors, in the order of a signal level, a wavekind level where the signal is configured by a plurality of wavekinds, a character level where the wavekind is configured by a plurality of characters.

The set of events derived from the original test language is stored in a table format having columns assigned to the data showing the number of subsequent edges and the starting value. The table storing the set of events is optimized by studying the starting value of a particular event based on an ending state produced by the previous event, thereby simplifying the data in the table.

Brief Description of the Drawings

Figure 1 is a diagram showing an example of format in STIL which describes signals or signal groups in the digital test vectors which constitute intended test vectors.

Figure 2 is a diagram showing an example of format in STIL which describes timings of edge in each of the signals constituting the waveforms.

Figure 3 is a diagram showing an example of format in STIL which describes patterns of vectors in each of the signals constituting the waveforms.

Figure 4 is a diagram showing an example of format in STIL which describes a flow of patterns in the test vectors constituting the intended waveforms.

Figure 5 is a diagram showing an example of format in TDL (Test Description Language) which is a test language native to ATE system developed by the assignee of this invention.

Figure 6A is a schematic diagram showing a basic principle of test language conversion in accordance with the present invention, and Figure 6B is a schematic diagram showing an example of functional configuration in the test language conversion of the present invention.

Figure 7 is a diagram showing an example of STIL construct and an equivalent TDL representation, and a template of waveform based on the TDL representation.

Figure 8 is a diagram showing other examples of STIL constructs, waveforms corresponding to the STIL constructs, and the TDL representations.

Figure 9 is a diagram showing further examples of STIL constructs, waveforms corresponding to the STIL constructs, and the TDL representations for explaining the optimization waveforms.

Figures 10A and 10B are tables showing examples of array depicting starting values and number of edges which represent decomposed events in the STIL test vectors for conducting the pattern match of the present invention.

Figure 11 is a flow diagram showing an example of procedure in the wavekind matching of the present invention in the different levels of test vectors.

Figure 12 is a flow diagram showing an example of procedure in the wavekind matching of the present invention from the previous cycle to the next cycle.

Figure 13 is a diagram showing a basic idea of converting the STIL test vectors to multi-clock test signals of TDL when types of DUT pin are appropriate to such multi-

clock signals.

Figure 14 is a waveform diagram showing a basic idea of converting the STIL test vectors to the pin multiplexing test vectors of TDL.

Detailed Descriptions of the Preferred Embodiments

The STIL (Standard Test Interface Language) testing language defines pattern and timing information for device testing using a cycle-based representation. The format used does not represent the language of any ATE systems, but has been constructed to provide the maximum flexibility in designing test programs. STIL has only recently been adopted as a standard and is not yet in widespread use. A number of STIL-based applications have been proposed in recent years, including general test flow, scan test methodology and ATPG (automatic test pattern generator). Currently, most ATE systems do not use STIL as a native language. Consequently, there is a need to convert from STIL to the native language of the target ATE system. This language is usually cycle-based, so the conversion is from one cycle-based representation to another.

In this patent specification, the inventor discloses some of the basic steps involved in conversion method from one cycle-based format to another cycle-based format. The inventor also presents some tricks that allow the conversion of STIL in an efficient manner. The conversions shown in this invention are based on an analysis of the data presented in STIL. As a sample target language, Test Description Language (TDL) by developed by Advantest Corporation, Tokyo, Japan, an assignee of this invention, is used for illustration purposes although the principles presented are general and thus equally applicable to other test languages.

Figures 1-4 showing examples of format in STIL for describing the digital test vectors. Figure 1 shows formats of signals and signal groups, Figure 2 shows formats of timings of edges in each signal. Figure 3 shows an example

of STIL format describing patterns of test vectors, and Figure 4 shows an example of STIL format describing the flow of patterns. Figure 5 is a diagram showing an example of format in TDL which is a target test language.

Template Matching

The conversion from one cycle-based representation to another is most easily accomplished using an event-based intermediate format. Here, event is any change such as edges in test vectors or no changes defined relative to timing. The two cycle-based descriptions will usually be substantially different from one another. For converting one cycle-based description to another cycle-based description, an intervening event-based representation is used to decompose the data to basic building blocks. Namely, the input cycle-based format (STIL) is decomposed into constituent events and these events are reconstructed in the target description format (TDL).

This basic process is illustrated in Figure 6A which performs the vector-based to vector-based conversion process. In the present invention, the test vectors in the STIL format is decomposed into each event which is compared with templates produced based on waveforms defined in TDL, an example of target test language. This concept is shown by the arrows of dotted lines in Figure 6A. When match is found for a template, the matched template is listed in a file to which the parameters in the corresponding STIL test vectors are transferred to complete the waveform. In this manner, the TDL test vectors are created as shown by the arrow of dotted line in Figure 6A.

Figure 6B is a functional representation of the test vector conversion of the present invention from the STIL test language to the TDL test language. A STIL vector file 21 is a file storing STIL test vectors to be converted to TDL vectors through the conversion process of the present invention. Typically, the STIL test vectors in the STIL file

21 are derived from the design stage of semiconductor devices, i.e., CAE (Computer-Aided Engineering) environment or EDA (Electronics Design Automation) environment as a result of conducting logic simulation. The STIL vectors are decomposed into constituent events and stored in a decomposed event file 24.

As noted above, TDL is a test language developed by Advantest Corporation, the assignee of the present invention, to establish a logic test pattern (LPAT) file. The format of TDL is stored in a TDL wavekind file 22. Each waveform defined in the target test language TDL, is converted to a corresponding template having a set of components. Such templates of waveforms are stored in a template file 25.

A pattern matching is performed by a comparator 26 between the events from the event file 24 and the templates from the template file 25. When match is found, the waveform corresponding to TDL is listed in a file 28 which stores template matched data representation. Further, the details of parameters for the vectors of the matched template are transferred from the STIL vectors to the TDL vectors. Such details of the vectors include timings, pattern characters, and types of edge. Thus, in a TDL & LPAT file 29, the TDL test vectors corresponding to the STIL test vectors are created through the conversion process in the foregoing.

As an example, with reference to Figures 7 and 8, consider the following STIL construct, an equivalent TDL representation and the corresponding waveforms:

01 {'400ns' D/U;} => NRZ; T1=400ns; T2=400ns

where the STIl construct 01 {'400ns' D/U;} in the left means that the character '0' specifies a falling edge (D) at 400ns while the character '1' specifies a rising edge (U) at 400ns. This STIl representation corresponds to the waveforms shown in the upper part of Figure 8 and to a non-return zero (NRZ) waveform defined in TDL. In Figure 8, the waveform of character '0' shows the shaded portion from the start edge of

a test cycle to the falling edge T1 (400ns from the start edge. The shaded portion means that the logic state in this area is undefined. Thus, the character '0' defines the falling edge T1 at 400ns from the start of the test cycle whatever the current logic status is. Similarly, the waveform of the character '1' defines the falling edge T1 at 400ns from the start of the test cycle whatever the current logic status is. As noted above, this example of STIL construct corresponds to the NRZ waveform of TDL.

Therefore, in the conversion method of the present invention, the construction of TDL waveforms from the decomposed STIL waveform descriptions is carried out using a template matching method. The waveforms available for a given tester are read in at run time and stored in a manner that makes template matching easy. Thus, as a first step, a list of templates is established for each waveform such as NRZ defined in TDL. The template is characterized by the set of {pattern character, starting value, number of subsequent edges} pairs that describe the waveform. The representation of the NRZ waveform in the above example is shown in the lower part of Figure 7.

In the example of template in Figure 7, the first element on each line is the pattern character while the next two elements represent the {starting value, number of subsequent edges} pairs while the remaining entries indicate the names for the specified edges. Using this, the example above can be represented using the NRZ waveform with the following values:

01 => {0,0,0}, {0,1,1 400ns D}, {1,0,1 400ns U}, {1,1,0}

where the {pattern character, starting value, number of subsequent edges} are shown. Storage of all waveforms defined in the target test language, in this example TDL, in this manner allows a simple comparison of the decomposed STIL waveforms against the capabilities of a given waveform. Thus, the library of templates is prepared in the template

file 25 of Figure 6B.

In the comparison, the decomposed STIL waveform and the template of TDL are compared with one another. The STIL waveform is decomposed into each event which is expressed by a combination of starting value and number of edges. The data describing the starting value and number of edges of each event in the STIL waveform is stored in the event file 24 of Figure 6B. Thus, the comparison between the decomposed events and the template is made by comparing the starting value and the number of edges. This is done as a query against the waveforms, essentially asking, "can you support this many edges with this starting value?".

It should be noted that the mapping of an STIL waveform to a template requires matching all of the resulting triples that comprise the waveform. In the example above, the characters '01' from STIL result in the four triples shown in the template of Figure 7. All four of these must be mapped in order for these characters to be fully represented. For this example, the NRZ is capable of supporting all of the required triples.

As another example of basic template matching, consider the following STIL characters, TDL representation and waveforms with reference to Figure 8:

0 {} 1 {'200ns' U; '400ns' D;} => RZ; T1=200ns; T2=400ns;

which corresponds to a return zero (RZ) waveform in TDL the waveforms of which are shown in the center of Figure 8. Thus, the template of RZ waveform of TDL is set in the template file 25 of Figure 6B.

The template matching examples shown in the preceding section are very simple. They match directly with standard waveforms, NRZ and RZ. These waveforms may be used on the test equipment running with TDL without any resource penalty. However, in the real test implementation, more complex waveforms are also used. Consider the following example:

01 {'100ns' D; '200ns' D/U; '400ns' D;} with the waveforms shown in the lower part of Figure 8 and an upper half of Figure 9. This example means that the character '0' specifies a falling edge at 100ns, a falling edge at 200ns and a falling edge at 400ns wherein the last two edges are redundant and will not actually be retained. The character '1' specifies a falling edge at 100ns, a rising edge at 200ns and a falling edge at 400ns.

The pattern character "0" can be matched by a NRZ or RZ waveform. The pattern character "1" would require a more complex waveform, such as the XOR (exclusive OR) with values T3=100ns, T1=200ns, T2=400ns. There are several reasons why this is an undesirable situation. First, if these two characters are actually used in a pattern block (and their definition makes this likely), this will result in an on-the-fly wavekind switch. On many ATE systems, this results in a limiting of available resources. Also, on some systems, the use of the XOR waveform in certain situations can result in a reduction in tester resources. Clearly this is not desirable solution to this waveform matching problem. Thus, in the following, some tricks will be detailed that can be used to simplify complex match cases.

<u>Unused Initial Value</u>

In STIL, all of the information about the waveforms that will be used is presented up front in the WaveformTable construct. Analysis can be performed on this information to determine characteristics of the available wave shapes. This information can be used to make informed optimization. In this section, the concept of "unused initial value" is examined to see how this can help with optimization.

First, it should be noted that a set of pattern characters presented to a given signal defines a continuous waveform despite the use of the discrete characters. Consequently, the starting state experienced by a given character is based on the ending state produced by the

previous character. The set of ending value is, therefore, contained within the information in the WaveformTables. The exception to this statement is the signal starting value, which can be set arbitrarily by the user. In order to allow optimization of the template matching algorithm, it is imperative that the user make intelligent choices about starting value.

Referring back to the example shown above, and repeated here: 01 {'100ns' D; '200ns' D/U; '400ns' D;}, and the corresponding waveforms are shown in Figures 8 and 9. It has been already shown that a complete treatment of this set of characters results in undesirable consequences. In analyzing this set of characters, it is known that neither of them ends with a "U" (rising) value. Consequently, by initializing the signal with a "D" (falling) value, and these are the only available characters, the characters never begin any cycle in a "U" state. The waveform description shown in the lower left of Figure 9 reduces (almost) to that shown in the lower right of Figure 9. Thus, the simple RZ waveform (T1=200ns; T2=400ns) can be used to match this set of characters.

Table Based Analysis

With the concept of unused initial value, the mechanism of WaveformTable-based analysis of the STIL characters is described with reference to the tables of Figures 10A and 10B and charts of Figures 11 and 12. This entails cataloging the requirements of all pattern characters and forming composite representations of these data at several levels (S11 in Figure 11). Note that these data need not be compiled across all pins or pin groups, but only within a signal. For each signal, the wave shape requirement data are compiled at three levels (S12 in Figure 11):

1. Entire signal
2. WaveformTable, and
3. Individual pattern character.

Ideally, it would be desirable to characterize the

behavior of an entire signal using a single wavekind, so the data are compiled at this level. Failing this, the next logical level is the WaveformTable. It is assumed that switching WaveformTables will not happen often during the patterns, very possibly only in going from one pattern block to another. Furthermore, if the switching between pattern blocks occurs at the level of PatternExec blocks, these will be relegated to different tests in the TDL code and any differences in wavekinds will not result in an on-the-fly switch.

Finally, if the behaviors of the pattern characters within a WaveformTable cannot be represented by a single wavekind, the matching will be attempted on an individual character level. If this fails, it means that some pattern character in the STIL file contains requirements that cannot be satisfied by the test language of the target ATE system. In some cases, more advanced features of the target ATE system can be used to alleviate these problems. In other cases, this is simply reported as a fatal error in the conversion process.

The data to be stored at each of the levels indicated above is the same: the starting value and number of edges required by the STIL pattern characters. This is stored in an array whose dimensions are based on characteristics of the target ATE system. With use of the "unused initial value" technique described above, the capacity of array can be significantly reduced. The number of drive edges that can be supported per time set is read in at run time and this is used for one dimension of the array. The other dimension is two, the number of possible starting values "1" or "0" in a binary logic system. Each combination of starting value and number of edges that requires support is set to true (T). The rest are false (F).

Thus, for the above example:

```
0 {} 1{'200ns' U; '400ns' D;} => RZO; T1=200ns; T2=400ns
```

the array would look like the table shown in Figure 10A where it is assumed that at most four drive edges may be supported per time set. For another example:

23{'100ns' D; '200ns' D/U; '400ns' D;}

the table of Figure 10B now contains an additional entry. This table describes all of the information about these four waveform characters. The technique of the unused initial value noted above can be applied to this table because "1" is the initial value that will never occur (unless set during initialization). As a result, the entire column labeled "1" can be set to false, which results in the same situation as described previously with reference to the waveforms in the lower left and right of Figure 9. This mechanism can be applied to tables created at any of the levels, i.e, signal, waveform table, and pattern character, discussed .

The more detailed description regarding the template matching procedure is given here. As described above, the array (matcharray) of decomposed events each forming a starting value and number of edges is created in the event file 24 of Figure 6B to be compared with the counterpart waveform data in the template file 25. Once the "matcharray" described above has been created, and then reduced through the analysis of "unused initial values", the match against the available waveforms is made (S13 in Figure 11). As discussed above, the waveforms available to the target ATE system (read in at run time) are stored based on the starting value and number of edges they can support in the template file 25. The structure in the templates is analogous to the matcharray shown above except that the entry in each box of tables in Figures 10A and 10B is a collection of wavekind object pointers that can satisfy the indicated combination of starting value and number of edges.

It should be noted that a given wavekind will appear in several of these lists as they can support a variety of {starting value, number of edge} pairs. For example, the RZO

waveform would be found in the {0,0}, {0,2}, {1,0} and {1,1} array elements as it can support all of these combinations. Note that this refers to the RZO waveform in TDL where there is no activity for the pattern character '0'. Thus, the {1,0} combination exists. The industry standard RZ waveform would return to zero for both pattern characters '0' and '1', and the {1,0} combination would not be possible.

The wavekind object can be queried for all combinations that it can support. For example, for accessing the RZO object through the {0,0} entry, the matching process can simply ask the RZO object about other combinations "can you support {0,2}" which would return true, "can you support {1,2}" which would return false. Thus, the matching process would find the object via the numerically lowest {starting value, number of edge} pair that was required and then query the object for the rest of the desired states. If any of these queries fail, this wavekind will not work and the matching process goes to the next entry in the original list and try the process again.

Once a successful match has been made for a given wavekind, the values for parameters must be set (S14 in Figure 11). The decomposed STIL characters contain time values for transitions, and the wavekind objects contain strings for the names of their relevant resources. For the case of the RZO waveform, the match of the {0,2} case completes with the association of the time value 200ns with the string "T1", and 400ns with string "T2". These pairings are stored in a table corresponding to the STIL character '1' as defined in the WaveformTable.

The WaveformTable-based analysis method takes advantage of economies available when the source format is cycle-based. The information regarding the formats that will be used for the entire process is available up-front and can be processed, analyzed and optimized before pattern conversion begins. Once this has been completed, the processing of the

vectors becomes very simple. The STIL pattern character is accessed (S21 in Figure 12), the previous signal value (stored at the end of the previous cycle) is recalled (S22 in Figure 12) and these data are used to access the information stored for the {starting value, STIL pattern character} pair, i.e. the wavekind and parameter information determined from the matching process (S23 and S24 in Figure 12). Then, the ending value is stored for beginning of the next cycle (S25 in Figure 12).

This is markedly different from the situation where the source format is event-based, for example, VCD (Value Change Dump) data by Verilog. In this case, the wave format information is usually not available up front.. Matching of source data to target wave shapes must be done as the vectors are processed. This can lead to poor choices that are avoided using proper table analysis techniques.

Assuming the following two cycles in a waveform: If the template matching were done on a cycle-by-cycle basis (as would be the normal case without a table-based analysis), the first cycle would likely end up being mapped to NRZ; this is the simplest waveform that can satisfy the constraints. The second cycle is clearly RZO. If it were known about the second cycle when processing the first, it would have noted that the first cycle can also be mapped to RZO, and would have prevented an on-the-fly wavekind switch. A table-based analysis approach provides just this capability.

<u>Output Signal Conversion</u>

The preceding description has focused on the mapping of input STIL characters to TDL equivalents. The "input" here means a test pattern supplied (drive) to a pin of device under test (DUT). As previously noted, the test vectors include the test patterns (input) and strobes (output or compare). A strobe is a timing pulse either with edge (no pulse width) or window (predetermined pulse width) to sample the device output signal. Here, the output mapping

strategies are described which deals with conversion of strobe signals (compare) in the STIL format to the TDL format. The behaviors specified in the STIL file can be directly translated to pattern characters. For example:

LHZX {'0ns' Z; '0ns' X; '260ns' L/H/T/X;}

can be mapped directly to edge strobes for the appropriate value, if all are applicable. In other words, the template matching is unnecessary in this process. Similarly:

LHZX {'0ns' Z; '0ns' X; '260ns' l/h/t/x; '500ns' x}

can be mapped to the appropriate window strobe.

There are several points worth noting about the output conversion. The first is that some ATE may not allow switching of strobe type on-the-fly. Thus, edge and window strobes may not be mixed. A solution in this case is to make all strobes into window strobes, with edge strobes made into window strobes as narrow as allowed by the system. The second point is that STIL pattern characters that request multiple strobes within a cycle may or may not be compatible with the target ATE system. The capabilities of the ATE system family is built into the conversion tool with a resource file read at run time to indicate which subset of these features is actually present on a given target ATE system. This would include behaviors such as transition and double strobe modes.

Bidirectional Signal Conversion

Bidirectional signal conversion provides the most complexity in the entire process. Here, the term "bidirectional" means conversion of test language from the STIL formate to the native language format for the test vectors assigned to device pin which functions both input and output. All of the features required for input and output matching noted above are present as well as additional features unique to bidirectional signals. In addition, bidirectional signals often place limitations on tester resources above and beyond those placed by simple input or

output signals. As an example, the number of available edges may be reduced due to the need for driver control signals to determine the directional state of the bidirectional signal.

The considerations for driver control are based on the characteristics of the STIL pattern characters and the capabilities of the target ATE system. A standard paradigm is to provide two driver enable modes, one that mimics the NRZ behavior, and one for RZ. In the former case, the cycle becomes "drive" (input) at some point and remains that way through the end of the cycle. For the RZ, the cycle becomes "drive" and then "compare" (output) during the cycle. Note that the cycle being in "compare" mode does not mean that a comparison is actually taking place, just that the pin is to be treated as an output. This distinction becomes important with regard to target ATE capabilities for drive and compare in a cycle.

The driver enable mode is determined from the STIL pattern characteristics by noting that an NRZ driver enable mode is preferred since it requires fewer tester resources. This mode is chosen unless specifically required to use the RZ mode. This only happens when a "drive" region is surrounded by "compare" regions in the same cycle. Again, actual comparisons may not be occurring, but the device pin is acting as an output. The time values for the driver control edges are determined from the transition times for the signal direction.

The inclusion of driver type information makes the "matcharray" process described above quite a bit more complex. The drive portions of the signal are matched against the available waveforms while the overall character of a cycle, in terms of "drive" and "compare" portions are compared against the capabilities of the target ATE system.

STIL contains the concept of the "DrivePrior" event. This is meant to contain the most-recently-used drive value on a system. For input signals, the prior drive value is

always the last state of the signal, so this need not be considered. Output signals have no drive states, so this is irrelevant. Thus, the DrivePrior is only relevant for bidirectional signals where it represents the last drive state attained by the signal, regardless of any intervening strobe activity. On some ATE systems, the presence of a strobe character affects the state of the driver in the system. For example, if the last "drive" state is "D", and this is followed by a strobe for "H", the driver can be set to a "U" state for the next drive cycle. The "DrivePrior" is intended to handle this case. With regard to the conversion process, this prior state must be kept as well as the actual current state of the driver. The result of this is that the "matcharray" shown above contains four columns instead of two, corresponding to all possible pairs {previous value, DrivePrior}, i.e., the column heads (Figures 10A and 10B) are {0,0}, {0,1}, {1,0}, {1,1}.

The data used in these entries must be used properly in order for the concept of the "DrivePrior" to work properly. The desired initial state, as given by the "DrivePrior" value, must be reconciled with the actual state of the driver, previous value. Using the example above, the driver is in the "U" state, and the STIL pattern character is requesting the device to be in the "D" state based on the inclusion of the "DrivePrior" or "P" event in the description. The waveform chosen to match this pattern character must be able to reconcile these values, driving the pin from the "U" state to the "D" state by the time required. This can result in an additional edge not readily apparent from the STIL pattern character description.

<u>Special Features</u>

In the following, the test language conversion is further discussed as to the processing of special features provided in the target ATE system. These features represent those found on the assignee's ATE system, Advantest Model

T6600 IC tester family, but they are quite general and may be found, in some form, on a variety of test systems. Thus, brief discussion is made as to the algorithms used to map the STIL information to these features.

Multi-Clock Signals

The multi-clock (MCLK) signal type is commonly used for providing more pulses per cycle than the ATE system can theoretically provide. This is done for repetitive waveforms by essentially breaking the tester cycle into a series of subcycles (internally) and providing multiple copies of a basic waveform, one per subcycle. The result is the appearance of a greater number of edges than possible in the rated test cycle.

The key to the use of the MCLK paradigm is that the waveform must contain a basic repeatable unit within the cycle. During the template matching algorithm described above, the discovery that a STIL pattern character contains too many edges leads to an attempt to match the character using an MCLK format. For this to work, the number of edges must be even (the MCLK format is pulse-based). The constraints that must be satisfied for a single-pulse repeatable waveform are derived with regard to an upper waveform of Figure 13.

In order to have a repeatable unit, it is required that the widths of all pulses be the same. Furthermore, the spacing between the pulses must be equal to the sum of the space (A) before the first pulse plus the space (B) at the end of the last pulse, as shown. This means that it is necessary to have a basic repeatable unit that looks like RZ with T1=A, T2= A+PW, and a subcycle length of A+PW+B. If any of these conditions are not met it is unable to provide a single pulse basic repeatable unit.

It is possible to create a double-pulse basic repeatable unit. The requirements are derived based on the lower waveform of Figure 13. In this case there are two pulses per

repeatable block, and it requires that the corresponding pulses be the same width, in this case the pulses labeled PW1 and PW2. The sum of the begin and end space must match the interval between the repeatable unit repetitions, as above (repeatable unit space = A+B). Also, the spacing (C) between the pulses must be the same for all repeatable units. Thus, in this case, a subcycle length is A+PW1+C+PW2+B.

Pin Multiplexing

In the pin multiplexing (PMUX), two tester channels are combined to drive a single pin. This allows the resources for both tester channels to be used for the same signal, enabling drive (input) and compare (output) in the same cycle on ATE systems where this might not otherwise be available.

While the use of PMUX can make the ATE programming task easier, it adds complication to the conversion process by providing an additional degree of freedom. While this might seem desirable in that it provides flexibility, it makes the search of the wavekind space a little harder because requirements are not precisely pinned down. In the following, discussion will be made as to the implications of PMUX on the various pin types.

(1) Input PMUX

Input pin mapping is performed by a search of the wavekind space attempting to match the capabilities of the wavekind with the requirements of the STIL pattern character. The details of this approach have been discussed above. When an input signal is used with PMUX, the responsibility for matching the requirements of the STIL pattern character is shared between the two tester channels. The issue presented by the added flexibility is one of determining a "requirement sharing" methodology.

One approach taken is that the simplest sharing algorithm is not to share at all. If possible, one pin does all the work and the other does none. To this end, an attempt is made to map all of the edges specified by the STIL

pattern character into the first tester channel. The matching that takes place here is identical to that described above for non-PMUX input signals. The difference occurs if the match attempt fails, i.e. there is no wavekind that can satisfy some subset of a STIL pattern character's requirements. Previously, this situation either resulted in an attempt to use MCLK, or the reporting of an error. With the PMUX case, one edge is shifted from the first channel to the second, and the match is attempted for the first channel and second channel separately. The shifting of edges is continued until a match is possible for both channels for all sets of requirements for the STIL pattern character.

Adding to the complication for this algorithm is the need to maintain continuity between the two component channel behaviors. A given channel "remembers" its final state from the last cycle. The state of the driver to the device, however, has been set by the signals from the other channel. For example, if channel 1 drives the pin to an "H" state, it will have "H" as its previous state. Now, suppose channel 2 drives the pin to a "L" state. When channel 1 drives again, it will "remember" being in the "H" state, but the driver will actually be in the "L" state because of the action of channel 2. Coordination between the two channels comprising the signal is required to prevent this situation.

As an example, consider the following signal of the uppermost waveform in Figure 14, where T_0 denotes a cycle boundary. This signal is very simple and appears to have a NRZ character for the first signal. In fact, the second signal apparently needs to do nothing except stay low. Consider the first pass solution to this problem, shown in Figure 14.

The signal proposed for Channel (Pin) 1 provides the rising edge at T_a, as desired (second waveform from the top in Figure 14). Since there are no edges in the second portion of the signal, Pin 2 need have no edges (third

waveform from the top in Figure 14). In the second cycle, Pin 1 remains high as there are no edges. In the second part of the cycle, there is a low-going edge. Since Pin 2 is already in the "L" state, no edge will be generated.

It is clear from this example that the simple approach will not work in this case. The signal will look exactly like the waveform on Pin 1 as Pin 2 has no edges. This has happened because the continuity across channels has not been maintained. Pin 2 needs to be aware that Pin 1 has ended in an "H" state so that subsequent edges on the system, in this case the transition to "L" at T_a in the second cycle, will be handled properly. This can be accomplished by replacing the waveform for Pin 2 above with that shown in the bottom of Figure 15.

Here the waveform for Pin 2 is brought to the "H" state at the split time. This will have no effect on the composite signal as the driver is already in the "H" state due to the action of Pin 1. It does, however, condition Pin 2 to be in a consistent state with Pin 1. When the low-going edge happens in the second cycle during the Pin 2 portion this will result in a low-going transition on Pin 2 at T_b as desired. This demonstrates the channel consistency problem that must be handled for PMUX signals.

(2) Output PMUX

As with normal pattern matching, the conversion of output signals in the presence of PMUX is simpler than input conversion. The cycle is split into two portions, and strobe edges are assigned to the two channels (pins) based on their position relative to the split time. For simplicity, the split time is chosen as the middle of the period. Strobe edges occurring before this time are assigned to the first channel, and those occurring after are assigned to the second. One notable exception to this rule is window strobes. These may not be split across the boundary between channels. Consequently, the split time is chosen so that the

entire window strobe falls within one of the channels.

(3) Bidirectional PMUX

The use of PMUX with bidirectional signals allows features that might not otherwise be available. This is a primary use for the PMUX construct, rather than for pure input or output signals. Probably the single most important use of the PMUX is to allow drive and compare within a cycle, if the target ATE system does not allow this. For cases where a STIL pattern character for a bidirectional signal specifies pure drive or compare behavior for a cycle, the processing is virtually identical to that described with respect to the input and output pin multiplexing above. When the behavior is mixed, the split time for dividing the cycle between the channels is based on the time of the direction switch. This leads to a very natural division of the responsibilities of the two channels. The concept of continuity across channels that we discussed for input signals applies here as well, but must also take into account the effects of intervening strobes.

As has been described above, according to the present invention, the test vectors in an original test language are converted to a target test language with high efficiency and high accuracy.

Although only a preferred embodiment is specifically illustrated and described herein, it will be appreciated that many modifications and variations of the present invention are possible in light of the above teachings and within the purview of the appended claims without departing the spirit and intended scope of the invention.

WHAT IS CLAIMED IS:

1. A method of converting test vectors in an original cycle based test language into a target cycle based test language, comprising the following steps of:

reading available waveforms defined in the target test language and forming a set of templates depicting the waveforms where each template corresponds to a waveform of the target test language and includes data showing at least a starting value of a segment of waveform and a number of subsequent edges in the waveform;

reading the test vectors of the original test language and decomposing a waveform in the test vectors in the original test language into a set of constituent events where each event includes data showing at least a starting value and a number of subsequent edges of the waveform;

comparing the template derived from the waveform in the target test language and the set of events derived from the original test language;

storing the waveform data in the target test language when a match is detected in the comparison step and retrieving corresponding parameters of the waveform in the test vectors of original test language and storing the parameters in combination with the matched waveform data;

repeating the above steps for all of the test waveforms in the original test language, thereby forming representation of the waveforms in the target test language.

2. A method of converting test vectors as defined in Claim 1, wherein the step of applying the comparison algorithm at different levels of abstraction, in the order of a signal level, a wavekind level where the signal is configured by a plurality of wavekinds, and a character level

where the wavekind is configured by a plurality of characters.

3. A method of converting test vectors as defined in Claim 1, wherein the set of events is stored in a table format having columns assigned to the data showing the number of subsequent edges and the starting value.

4. A method of converting test vectors as defined in Claim 3, wherein the table storing the set of events is optimized by studying the starting value of a particular event based on an ending state produced by the previous event, thereby simplifying the data in the table.

5. A method of converting test vectors as defined in Claim 1, wherein the test vectors includes drive signals to be supplied to a device under test (DUT) as an input and strobe signals to sample an output of DUT for evaluation, wherein the drive signals in the original test language are converted to the target test language by comparing the template and the set of events for detecting the match while the strobe signals in the original test language are directly translated to the target test language.

6. A method of converting test vectors as defined in Claim 1, wherein the waveforms in the original test language are assigned where required by resource limitations to a plurality of subcycles of the target test language where the plurality of subcycles are created by multiplexing a test cycle clock in a test system which is operated by the target test language.

7. A method of converting test vectors as defined in Claim 1, wherein the waveforms in the original test language are assigned to a plurality of test channels of the target test language where the plurality of test channels are multiplexed to be connected to a single pin of DUT in a manner configured by a test system which is operated by the target test language.

8. A method of converting test vectors in a STIL

(Standard Test Interface Language) into a target cycle based test language, comprising the following steps of:

reading available waveforms defined in the target test language and forming a set of templates depicting the waveforms where each template corresponds to a waveform of the target test language and includes data showing at least a starting value of a segment of waveform and a number of subsequent edges in the waveform;

reading the test waveforms of the STIL format and decomposing a waveform in the test vectors in the STIL format into a set of constituent events where each event includes data showing at least a starting value and a number of subsequent edges of the waveform;

comparing the template derived from the waveform in the target test language and the set of events derived from the waveform in STIL;

storing the waveform data in the target test language when a match is detected in the comparison step and retrieving corresponding parameters of the waveform in the test vectors of STIL and storing the parameters in combination with the matched waveform data;

repeating the above steps for all of the test vectors in STIL, thereby forming a test vector file of the target test language.

Fig. 1
Example of STIL(1)

Signals:

```
Signals {
    Signal[0] InOut;
    Signal[1] InOut;
    Signal[2] InOut;
}

SignalGroups {
    signal = 'Signal[0..2]';
}
```

Fig. 2
Example of STIL(2)

Timing:

```
Timing T1
{
    WaveformTable wft1
    {
        Period '1us';
        Waveforms
        {
            Signal[0] {01 { '400ns' D/U; }}
            Signal[1] {01 { '400ns' D/U; '500ns' D;}
                23 { '0ns' U/D; '200ns' D/U; '600ns' U/D;}}
            Signal[2] {01  { '200ns' D/U; }
                23 {'100ns' D/U; '400ns' D; }}
        } // end of Waveforms block
    } // End of WaveformTable wft1
} // End of Timing block
```

Fig. 3
Example of STIL(3)

Pattern:

```
Pattern pat1 {

        W wft1;

        V { signal  =011; } // Note: signal = Signal[0] +
        V { signal  =020; } // Signal[1] + Signal[2]
        V { signal  =111; }
        V { signal  =103; }
        V { signal  =132; }
        V { signal  =000; }
        V { signal  =003; }
}
```

Fig. 4
Example of STIL(4)

Flow:

```
PatternBurst pb1 {
    PatList {
        pat1;
    } // end of PatList
} // end of PatternBurst

PatternExec {
    Timing T1;
    PatternBurst pb1;
} // end of PatternExec
```

Fig. 5
Example of TDL

Timing and Pattern:

```
SIGNAL signal_1;
signal_1.drekind( 0, NRZ);
signal_1.wavekind( 1, RZO);
signal_1.timing( 1, T1, 400.0nS);
signal_1.timing( 1, T2, 500.0nS);
signal_1.wavekind( 2, XOR);
signal_1.timing( 2, T3, 0.0nS);
signal_1.timing( 2, T1, 200.0nS);
signal_1.timing( 2, T2, 600.0nS);
signal_1.wavekind( 3, NRZ);
signal_1.wavekind( 4, NRZ);
signal_1.timing( 4, STBL, 0.0nS);
signal_1.timing( 4, DREL, 0.0nS);
```

Fig. 6A
Conversion Basics

- Vector-based to vector-based conversion

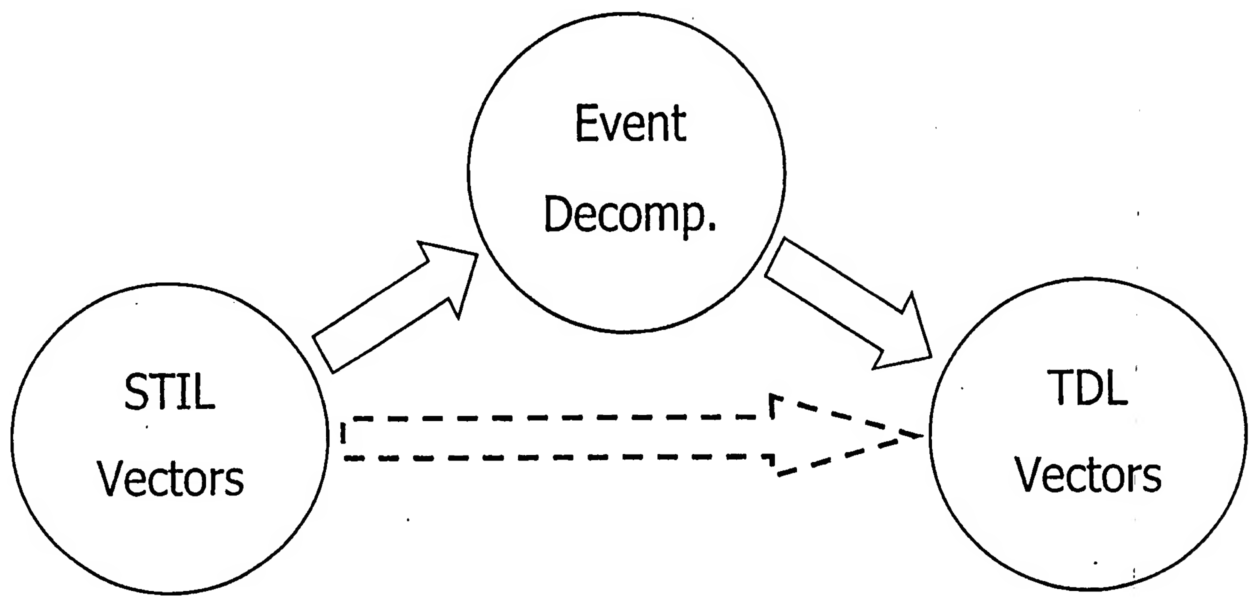


Fig. 6B

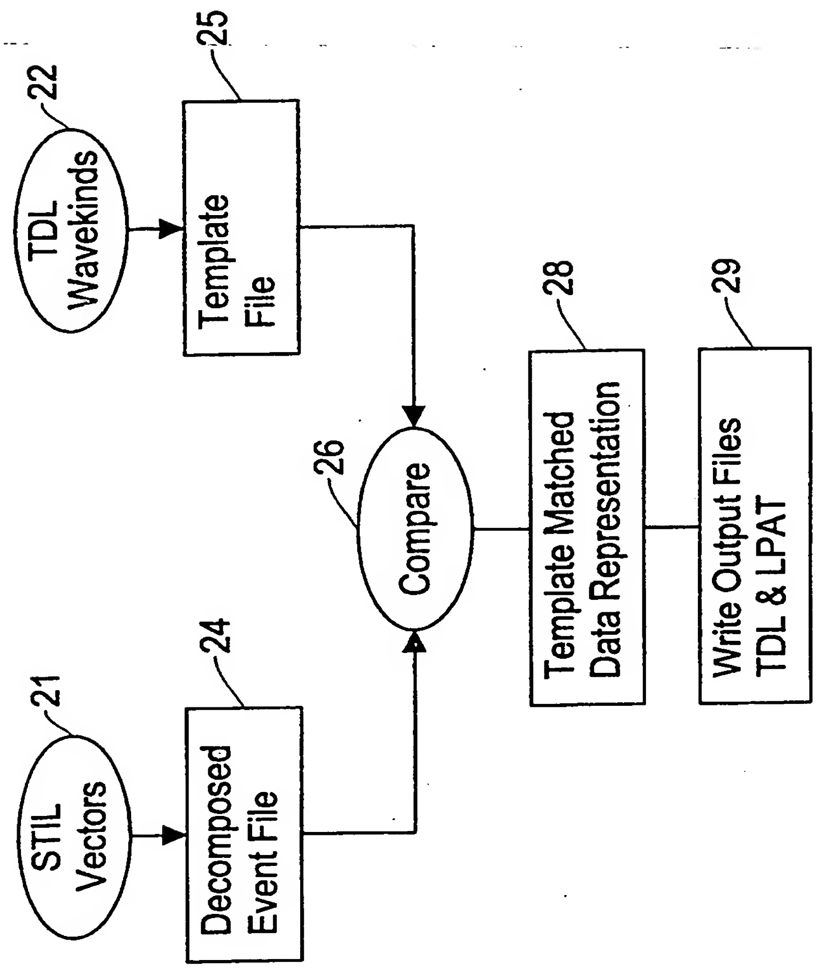

STIL Vectors
21
TDL Wavekinds
22
Decomposed Event File
24
Template File
25
Compare
26
Template Matched Data Representation
28
Write Output Files TDL & LPAT
29

Fig. 7
Template Matching

- 01 {'400ns' D/U;} => NRZ; T1 = 400ns; T2 = 400ns

- Template

```
{0,0,0}
{0,1,1 T2}
{1,0,1 T1}
{1,1,0}
```

Fig. 8
Wavekind Matching (1)

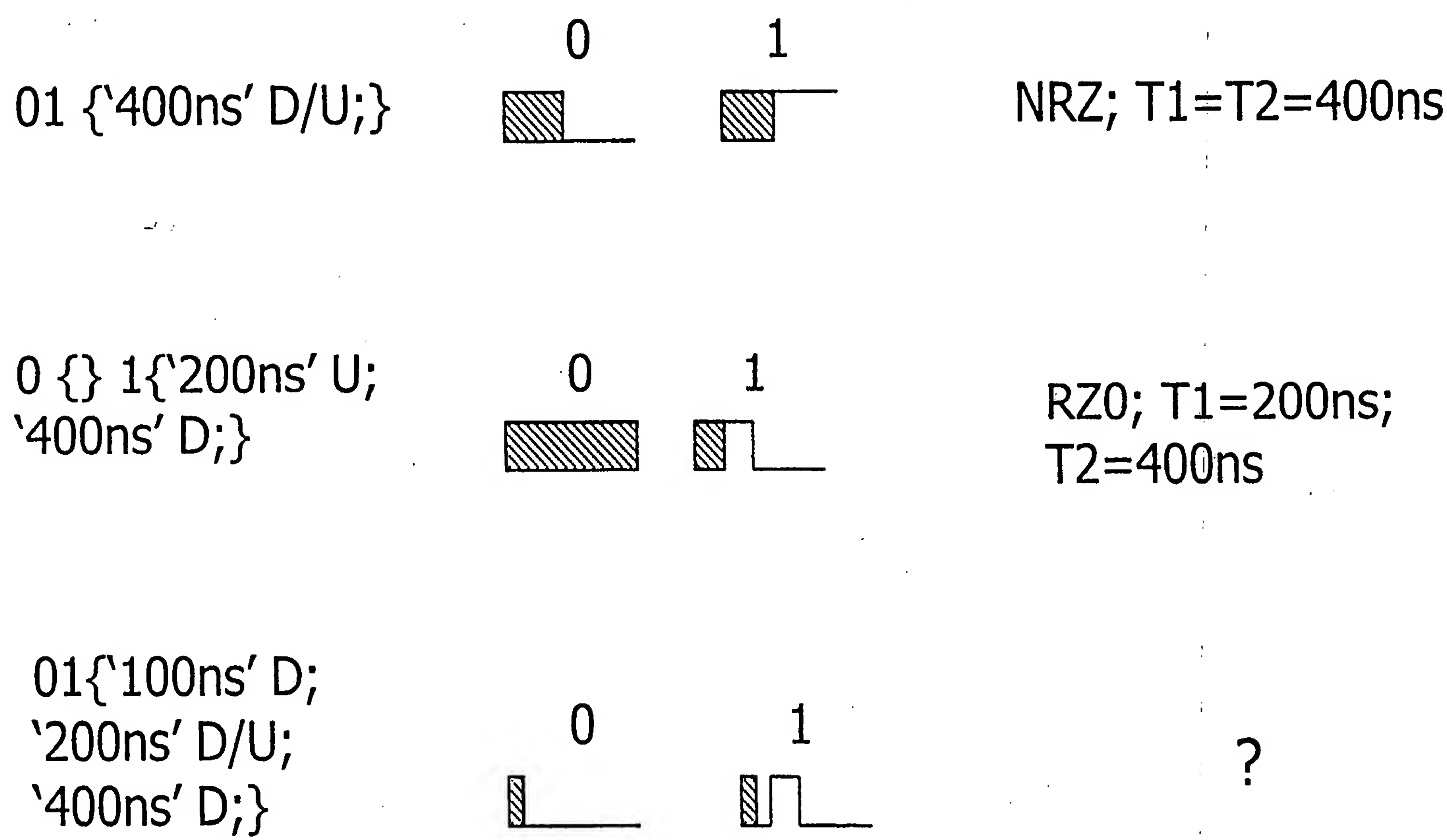
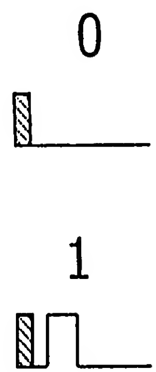


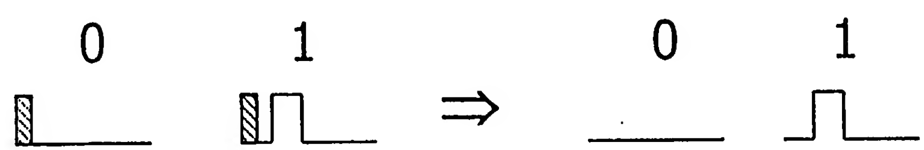
Fig. 9
Wavekind Matching (2)



NRZ; RZO...

XOR; T3=100ns; T1=200ns; T2=400ns

- Analyze timing block
  - What if no waveform can end in 'U' state?



RZ0; T1=200ns; T2=400ns

Fig. 10A

Number of Edges	Start Value	
	0	1
0	T	T
1	F	T
2	T	F
3	F	F
4	F	F

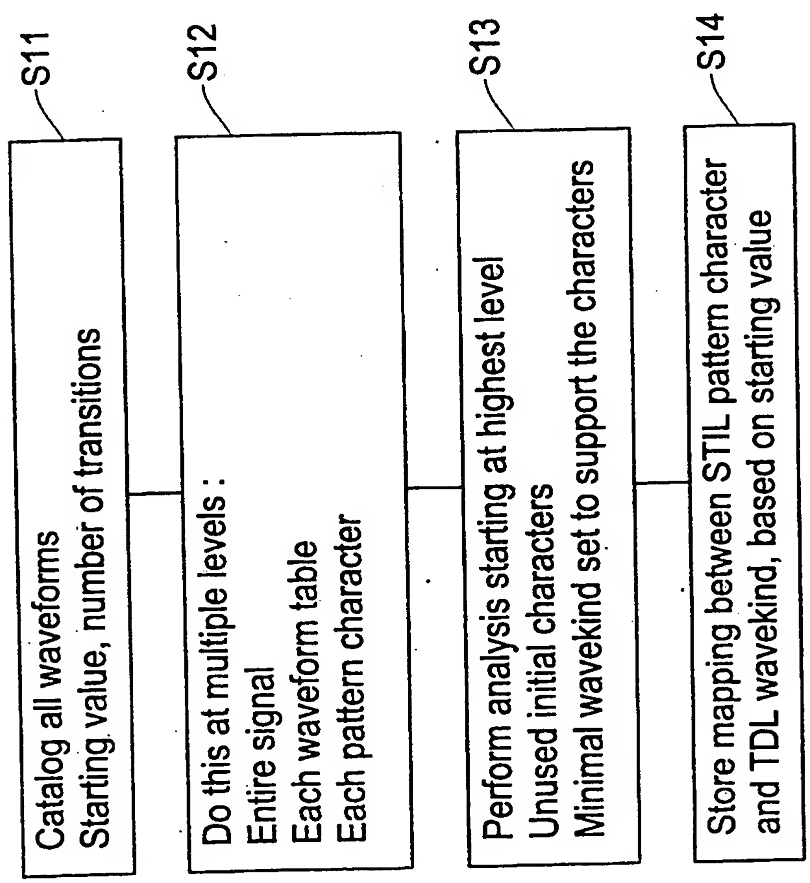
0{}1{'200ns' U:'400ns'D;}

Fig. 10B

Number of Edges	Start Value	
	0	1
0	T	T
1	F	T
2	T	F
3	F	T
4	F	F

23{'100ns' D; '200ns' D/U; '400ns' D;}

Fig. 11


S11
Catalog all waveforms
Starting value, number of transitions
S12
Do this at multiple levels :
Entire signal
Each waveform table
Each pattern character
S13
Perform analysis starting at highest level
Unused initial characters
Minimal wavekind set to support the characters
S14
Store mapping between STIL pattern character
and TDL wavekind, based on starting value

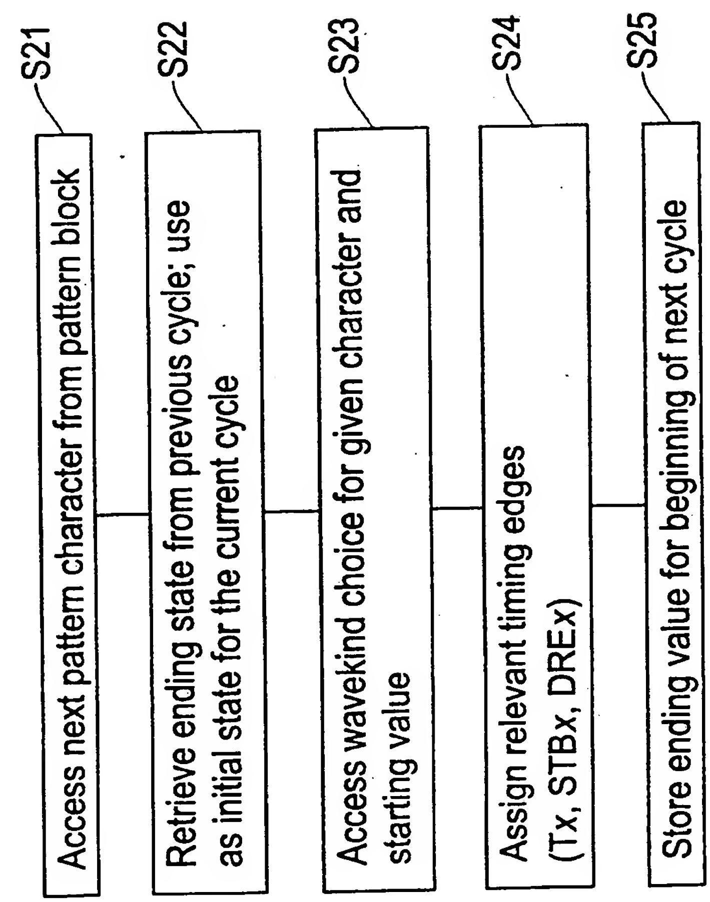
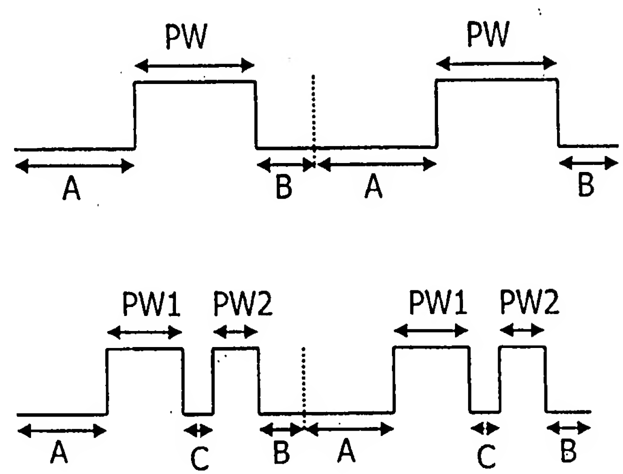

S21
Access next pattern character from pattern block
S22
Retrieve ending state from previous cycle; use as initial state for the current cycle
S23
Access wavekind choice for given character and starting value
S24
Assign relevant timing edges (Tx, STBx, DREx)
S25
Store ending value for beginning of next cycle

Fig. 12

Fig. 13
Multi-Clock (MCLK)

- MCLK
  - Find repetitive basic timing unit in STIL pattern char
  - Can be based on single- or double-pulse waveform



- Determine number of repetitions and factor into Rate

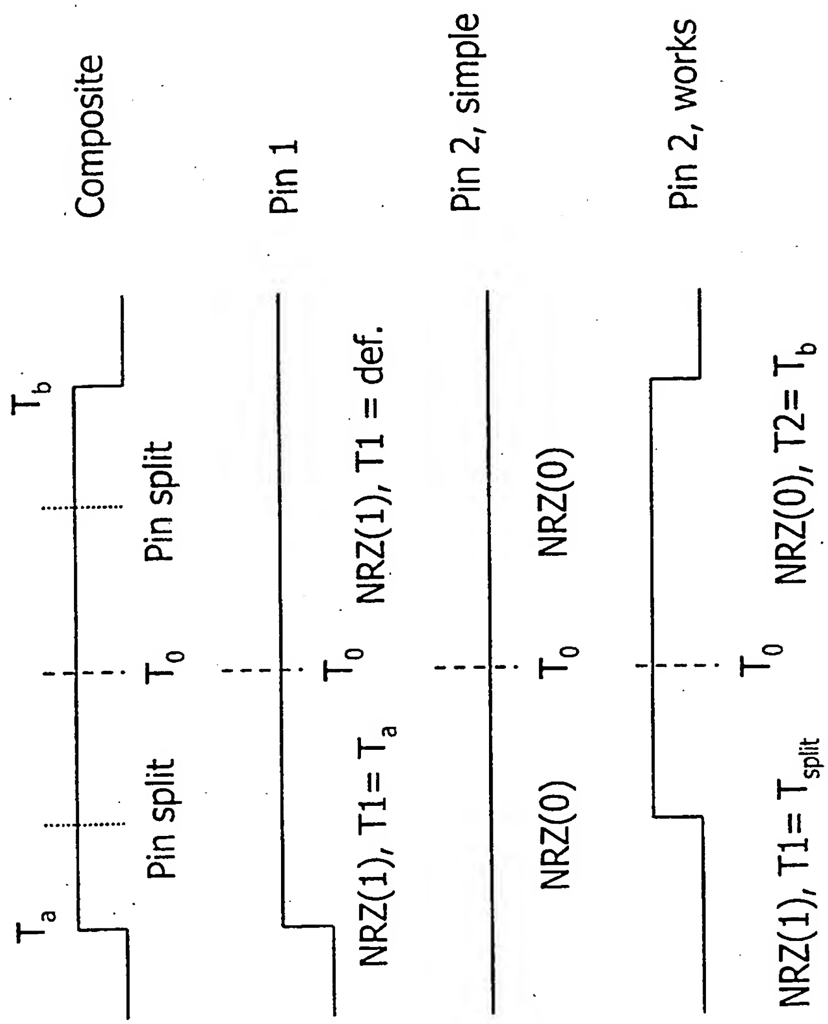


Fig. 14
Pin Mux

10/089137
JC13 Rec'd PCT/PTO 23 MAR 2002

15/prts

TEST LANGUAGE CONVERSION METHOD

Field of the Invention

This invention relates to a test data conversion method for testing semiconductor devices by automatic test equipment, and more particularly, to a method of converting digital test vectors written in STIL (Standard Test Interface Language) to digital test vectors of a test language unique to the automatic test equipment.

Background of the Invention

In testing semiconductor devices such as ICs and LSIs by automatic test equipment (ATE) or an IC tester, a semiconductor IC device to be tested is provided with test signals or test patterns produced by an IC tester at its appropriate pins at predetermined test timings. The IC tester receives output signals from the IC device under test in response to the test signals. The output signals are strobed or sampled by strobe signals generated by the IC tester with predetermined timings to be compared with expected data to determine whether the IC device functions correctly or not. Typically, timings of the test signals and strobe signals are defined relative to a start timing of each test cycle of IC tester.

As noted above, an IC tester generates test patterns and strobes, i.e., test vector, based on digital test vector data described in a language (format) unique to the tester. Such languages for automatic test equipment are different from manufacturer to manufacturer.

Recently, IEEE has proposed a test language STIL (Standard Test Interface Language) as a standard test interface language (IEEE Std 1450-1999). STIL provides an interface between computer-aided engineering (CAE) such as a logic test simulator and automatic test equipment. In a CAE environment or an EDA (electronic design automation)

environment, a semiconductor device is designed with the aid of computer system and such design is tested through a logic test simulator or a testbench. It is preferable to utilize the digital test vectors resulted from the logic simulation in testing actual semiconductor devices by IC testers. STIL is designed to facilitate the transfer of large volumes of digital test vectors from CAE environment to the automatic test equipment (ATE) environment.

The STIL test language has recently become a standard. At present, however, most ATE systems do not use STIL as a native language. Such native languages provided by test equipment manufacturers are not compatible to one another. Accordingly, there is a need to effectively convert STIL to the ATE native language.

Summary of the Invention

It is, therefore, an object of the present invention to provide a method of converting digital test vectors of one cycle based test language to another cycle based test language with high efficiency and accuracy.

It is another object of the present invention to provide a test language conversion method of converting test vectors in a STIL (Standard Test Interface Language) format into a target with high efficiency and accuracy.

In the present invention, the method of converting test vectors in an original cycle based test language into a target cycle based test language, comprising the following steps of: reading available waveforms defined in the target test language and forming a set of templates depicting the waveforms where each template corresponds to a waveform of the target test language and includes data showing at least a starting value of a segment of waveform and a number of subsequent edges in the waveform; reading the test vectors of the original test language and decomposing a waveform in the test vectors in the original test language into a set of

constituent events where each event includes data showing at least a starting value and a number of subsequent edges of the waveform; comparing the template derived from the waveform in the target test language and the set of events derived from the original test language; storing the waveform data in the target test language when a match is detected in the comparison step and retrieving corresponding parameters of the waveform in the test vectors of original test language and storing the parameters in combination with the matched waveform data; and repeating the above steps for all of the test vectors in the original test language, thereby forming a test vector file of the target test language.

Typically, the original test language is STIL (Standard Test Interface Language) specified by IEEE Std 1450-1999. Preferably, the step of comparing the template and the set of events includes a step of changing different levels of test vectors, in the order of a signal level, a wavekind level where the signal is configured by a plurality of wavekinds, a character level where the wavekind is configured by a plurality of characters.

The set of events derived from the original test language is stored in a table format having columns assigned to the data showing the number of subsequent edges and the starting value. The table storing the set of events is optimized by studying the starting value of a particular event based on an ending state produced by the previous event, thereby simplifying the data in the table.

Brief Description of the Drawings

Figure 1 is a diagram showing an example of format in STIL which describes signals or signal groups in the digital test vectors which constitute intended test vectors.

Figure 2 is a diagram showing an example of format in STIL which describes timings of edge in each of the signals constituting the waveforms.

Figure 3 is a diagram showing an example of format in STIL which describes patterns of vectors in each of the signals constituting the waveforms.

Figure 4 is a diagram showing an example of format in STIL which describes a flow of patterns in the test vectors constituting the intended waveforms.

Figure 5 is a diagram showing an example of format in TDL (Test Description Language) which is a test language native to ATE system developed by the assignee of this invention.

Figure 6A is a schematic diagram showing a basic principle of test language conversion in accordance with the present invention, and Figure 6B is a schematic diagram showing an example of functional configuration in the test language conversion of the present invention.

Figure 7 is a diagram showing an example of STIL construct and an equivalent TDL representation, and a template of waveform based on the TDL representation.

Figure 8 is a diagram showing other examples of STIL constructs, waveforms corresponding to the STIL constructs, and the TDL representations.

Figure 9 is a diagram showing further examples of STIL constructs, waveforms corresponding to the STIL constructs, and the TDL representations for explaining the optimization waveforms.

Figures 10A and 10B are tables showing examples of array depicting starting values and number of edges which represent decomposed events in the STIL test vectors for conducting the pattern match of the present invention.

Figure 11 is a flow diagram showing an example of procedure in the wavekind matching of the present invention in the different levels of test vectors.

Figure 12 is a flow diagram showing an example of procedure in the wavekind matching of the present invention

from the previous cycle to the next cycle.

Figure 13 is a diagram showing a basic idea of converting the STIL test vectors to multi-clock test signals of TDL when types of DUT pin are appropriate to such multi-clock signals.

Figure 14 is a waveform diagram showing a basic idea of converting the STIL test vectors to the pin multiplexing test vectors of TDL.

Detailed Descriptions of the Preferred Embodiments

The STIL (Standard Test Interface Language) testing language defines pattern and timing information for device testing using a cycle-based representation. The format used does not represent the language of any ATE systems, but has been constructed to provide the maximum flexibility in designing test programs. STIL has only recently been adopted as a standard and is not yet in widespread use. A number of STIL-based applications have been proposed in recent years, including general test flow, scan test methodology and ATPG (automatic test pattern generator). Currently, most ATE systems do not use STIL as a native language. Consequently, there is a need to convert from STIL to the native language of the target ATE system. This language is usually cycle-based, so the conversion is from one cycle-based representation to another.

In this patent specification, the inventor discloses some of the basic steps involved in conversion method from one cycle-based format to another cycle-based format. The inventor also presents some tricks that allow the conversion of STIL in an efficient manner. The conversions shown in this invention are based on an analysis of the data presented in STIL. As a sample target language, Test Description Language (TDL) by developed by Advantest Corporation, Tokyo, Japan, an assignee of this invention, is used for illustration purposes although the principles presented are

general and thus equally applicable to other test languages.

Figures 1-4 showing examples of format in STIL for describing the digital test vectors. Figure 1 shows formats of signals and signal groups, Figure 2 shows formats of timings of edges in each signal. Figure 3 shows an example of STIL format describing patterns of test vectors, and Figure 4 shows an example of STIL format describing the flow of patterns. Figure 5 is a diagram showing an example of format in TDL which is a target test language.

Template Matching

The conversion from one cycle-based representation to another is most easily accomplished using an event-based intermediate format. Here, event is any change such as edges in test vectors or no changes defined relative to timing. The two cycle-based descriptions will usually be substantially different from one another. For converting one cycle-based description to another cycle-based description, an intervening event-based representation is used to decompose the data to basic building blocks. Namely, the input cycle-based format (STIL) is decomposed into constituent events and these events are reconstructed in the target description format (TDL).

This basic process is illustrated in Figure 6A which performs the vector-based to vector-based conversion process. In the present invention, the test vectors in the STIL format is decomposed into each event which is compared with templates produced based on waveforms defined in TDL, an example of target test language. This concept is shown by the arrows of dotted lines in Figure 6A. When match is found for a template, the matched template is listed in a file to which the parameters in the corresponding STIL test vectors are transferred to complete the waveform. In this manner, the TDL test vectors are created as shown by the arrow of dotted line in Figure 6A.

Figure 6B is a functional representation of the test vector conversion of the present invention from the STIL test language to the TDL test language. A STIL vector file 21 is a file storing STIL test vectors to be converted to TDL vectors through the conversion process of the present invention. Typically, the STIL test vectors in the STIL file 21 are derived from the design stage of semiconductor devices, i.e., CAE (Computer-Aided Engineering) environment or EDA (Electronics Design Automation) environment as a result of conducting logic simulation. The STIL vectors are decomposed into constituent events and stored in a decomposed event file 24.

As noted above, TDL is a test language developed by Advantest Corporation, the assignee of the present invention, to establish a logic test pattern (LPAT) file. The format of TDL is stored in a TDL wavekind file 22. Each waveform defined in the target test language TDL, is converted to a corresponding template having a set of components. Such templates of waveforms are stored in a template file 25.

A pattern matching is performed by a comparator 26 between the events from the event file 24 and the templates from the template file 25. When match is found, the waveform corresponding to TDL is listed in a file 28 which stores template matched data representation. Further, the details of parameters for the vectors of the matched template are transferred from the STIL vectors to the TDL vectors. Such details of the vectors include timings, pattern characters, and types of edge. Thus, in a TDL & LPAT file 29, the TDL test vectors corresponding to the STIL test vectors are created through the conversion process in the foregoing.

As an example, with reference to Figures 7 and 8, consider the following STIL construct, an equivalent TDL representation and the corresponding waveforms:

01 {'400ns' D/U;} => NRZ; T1=400ns; T2=400ns

where the STIl construct 01 {'400ns' D/U;} in the left means that the character '0' specifies a falling edge (D) at 400ns while the character '1' specifies a rising edge (U) at 400ns. This STIl representation corresponds to the waveforms shown in the upper part of Figure 8 and to a non-return zero (NRZ) waveform defined in TDL. In Figure 8, the waveform of character '0' shows the shaded portion from the start edge of a test cycle to the falling edge T1 (400ns from the start edge. The shaded portion means that the logic state in this area is undefined. Thus, the character '0' defines the falling edge T1 at 400ns from the start of the test cycle whatever the current logic status is. Similarly, the waveform of the character '1' defines the falling edge T1 at 400ns from the start of the test cycle whatever the current logic status is. As noted above, this example of STIL construct corresponds to the NRZ waveform of TDL.

Therefore, in the conversion method of the present invention, the construction of TDL waveforms from the decomposed STIL waveform descriptions is carried out using a template matching method. The waveforms available for a given tester are read in at run time and stored in a manner that makes template matching easy. Thus, as a first step, a list of templates is established for each waveform such as NRZ defined in TDL. The template is characterized by the set of {pattern character, starting value, number of subsequent edges} pairs that describe the waveform. The representation of the NRZ waveform in the above example is shown in the lower part of Figure 7.

In the example of template in Figure 7, the first element on each line is the pattern character while the next two elements represent the {starting value, number of subsequent edges} pairs while the remaining entries indicate the names for the specified edges. Using this, the example above can be represented using the NRZ waveform with the

following values:

01 => {0,0,0}, {0,1,1 400ns D}, {1,0,1 400ns U}, {1,1,0} where the {pattern character, starting value, number of subsequent edges} are shown. Storage of all waveforms defined in the target test language, in this example TDL, in this manner allows a simple comparison of the decomposed STIL waveforms against the capabilities of a given waveform. Thus, the library of templates is prepared in the template file 25 of Figure 6B.

In the comparison, the decomposed STIL waveform and the template of TDL are compared with one another. The STIL waveform is decomposed into each event which is expressed by a combination of starting value and number of edges. The data describing the starting value and number of edges of each event in the STIL waveform is stored in the event file 24 of Figure 6B. Thus, the comparison between the decomposed events and the template is made by comparing the starting value and the number of edges. This is done as a query against the waveforms, essentially asking, "can you support this many edges with this starting value?".

It should be noted that the mapping of an STIL waveform to a template requires matching all of the resulting triples that comprise the waveform. In the example above, the characters '01' from STIL result in the four triples shown in the template of Figure 7. All four of these must be mapped in order for these characters to be fully represented. For this example, the NRZ is capable of supporting all of the required triples.

As another example of basic template matching, consider the following STIL characters, TDL representation and waveforms with reference to Figure 8:

0 {} 1 {'200ns' U; '400ns' D;} => RZ; T1=200ns; T2=400ns;

which corresponds to a return zero (RZ) waveform in TDL the

waveforms of which are shown in the center of Figure 8. Thus, the template of RZ waveform of TDL is set in the template file 25 of Figure 6B.

The template matching examples shown in the preceding section are very simple. They match directly with standard waveforms, NRZ and RZ. These waveforms may be used on the test equipment running with TDL without any resource penalty. However, in the real test implementation, more complex waveforms are also used. Consider the following example: 01 {'100ns' D; '200ns' D/U; '400ns' D;} with the waveforms shown in the lower part of Figure 8 and an upper half of Figure 9. This example means that the character '0' specifies a falling edge at 100ns, a falling edge at 200ns and a falling edge at 400ns wherein the last two edges are redundant and will not actually be retained. The character '1' specifies a falling edge at 100ns, a rising edge at 200ns and a falling edge at 400ns.

The pattern character "0" can be matched by a NRZ or RZ waveform. The pattern character "1" would require a more complex waveform, such as the XOR (exclusive OR) with values T3=100ns, T1=200ns, T2=400ns. There are several reasons why this is an undesirable situation. First, if these two characters are actually used in a pattern block (and their definition makes this likely), this will result in an on-the-fly wavekind switch. On many ATE systems, this results in a limiting of available resources. Also, on some systems, the use of the XOR waveform in certain situations can result in a reduction in tester resources. Clearly this is not desirable solution to this waveform matching problem. Thus, in the following, some tricks will be detailed that can be used to simplify complex match cases.

Unused Initial Value

In STIL, all of the information about the waveforms that will be used is presented up front in the WaveformTable

construct. Analysis can be performed on this information to determine characteristics of the available wave shapes. This information can be used to make informed optimization. In this section, the concept of "unused initial value" is examined to see how this can help with optimization.

First, it should be noted that a set of pattern characters presented to a given signal defines a continuous waveform despite the use of the discrete characters. Consequently, the starting state experienced by a given character is based on the ending state produced by the previous character. The set of ending value is, therefore, contained within the information in the WaveformTables. The exception to this statement is the signal starting value, which can be set arbitrarily by the user. In order to allow optimization of the template matching algorithm, it is imperative that the user make intelligent choices about starting value.

Referring back to the example shown above, and repeated here: 01 {'100ns' D; '200ns' D/U; '400ns' D;}, and the corresponding waveforms are shown in Figures 8 and 9. It has been already shown that a complete treatment of this set of characters results in undesirable consequences. In analyzing this set of characters, it is known that neither of them ends with a "U" (rising) value. Consequently, by initializing the signal with a "D" (falling) value, and these are the only available characters, the characters never begin any cycle in a "U" state. The waveform description shown in the lower left of Figure 9 reduces (almost) to that shown in the lower right of Figure 9. Thus, the simple RZ waveform (T1=200ns; T2=400ns) can be used to match this set of characters.

Table Based Analysis

With the concept of unused initial value, the mechanism of WaveformTable-based analysis of the STIL characters is described with reference to the tables of Figures 10A and 10B

and charts of Figures 11 and 12. This entails cataloging the requirements of all pattern characters and forming composite representations of these data at several levels (S11 in Figure 11). Note that these data need not be compiled across all pins or pin groups, but only within a signal. For each signal, the wave shape-requirement data are compiled at three levels (S12 in Figure 11):

1. Entire signal
2. WaveformTable, and
3. Individual pattern character.

Ideally, it would be desirable to characterize the behavior of an entire signal using a single wavekind, so the data are compiled at this level. Failing this, the next logical level is the WaveformTable. It is assumed that switching WaveformTables will not happen often during the patterns, very possibly only in going from one pattern block to another. Furthermore, if the switching between pattern blocks occurs at the level of PatternExec blocks, these will be relegated to different tests in the TDL code and any differences in wavekinds will not result in an on-the-fly switch.

Finally, if the behaviors of the pattern characters within a WaveformTable cannot be represented by a single wavekind, the matching will be attempted on an individual character level. If this fails, it means that some pattern character in the STIL file contains requirements that cannot be satisfied by the test language of the target ATE system. In some cases, more advanced features of the target ATE system can be used to alleviate these problems. In other cases, this is simply reported as a fatal error in the conversion process.

The data to be stored at each of the levels indicated above is the same: the starting value and number of edges required by the STIL pattern characters. This is stored in

an array whose dimensions are based on characteristics of the target ATE system. With use of the "unused initial value" technique described above, the capacity of array can be significantly reduced. The number of drive edges that can be supported per time set is read in at run time and this is used for one dimension of the array. The other dimension is two, the number of possible starting values "1" or "0" in a binary logic system. Each combination of starting value and number of edges that requires support is set to true (T). The rest are false (F).

Thus, for the above example:

0 {} 1{'200ns' U; '400ns' D;} => RZO; T1=200ns; T2=400ns

the array would look like the table shown in Figure 10A where it is assumed that at most four drive edges may be supported per time set. For another example:

23{'100ns' D; '200ns' D/U; '400ns' D;}

the table of Figure 10B now contains an additional entry. This table describes all of the information about these four waveform characters. The technique of the unused initial value noted above can be applied to this table because "1" is the initial value that will never occur (unless set during initialization). As a result, the entire column labeled "1" can be set to false, which results in the same situation as described previously with reference to the waveforms in the lower left and right of Figure 9. This mechanism can be applied to tables created at any of the levels, i.e, signal, waveform table, and pattern character, discussed .

The more detailed description regarding the template matching procedure is given here. As described above, the array (matcharray) of decomposed events each forming a starting value and number of edges is created in the event file 24 of Figure 6B to be compared with the counterpart waveform data in the template file 25. Once the "matcharray" described above has been created, and then reduced through

the analysis of "unused initial values", the match against the available waveforms is made (S13 in Figure 11). As discussed above, the waveforms available to the target ATE system (read in at run time) are stored based on the starting value and number of edges they can support in the template file 25. The structure in the templates is analogous to the matcharray shown above except that the entry in each box of tables in Figures 10A and 10B is a collection of wavekind object pointers that can satisfy the indicated combination of starting value and number of edges.

It should be noted that a given wavekind will appear in several of these lists as they can support a variety of {starting value, number of edge} pairs. For example, the RZO waveform would be found in the {0,0}, {0,2}, {1,0} and {1,1} array elements as it can support all of these combinations. Note that this refers to the RZO waveform in TDL where there is no activity for the pattern character '0'. Thus, the {1,0} combination exists. The industry standard RZ waveform would return to zero for both pattern characters '0' and '1', and the {1,0} combination would not be possible.

The wavekind object can be queried for all combinations that it can support. For example, for accessing the RZO object through the {0,0} entry, the matching process can simply ask the RZO object about other combinations "can you support {0,2}" which would return true, "can you support {1,2}" which would return false. Thus, the matching process would find the object via the numerically lowest {starting value, number of edge} pair that was required and then query the object for the rest of the desired states. If any of these queries fail, this wavekind will not work and the matching process goes to the next entry in the original list and try the process again.

Once a successful match has been made for a given wavekind, the values for parameters must be set (S14 in

Figure 11). The decomposed STIL characters contain time values for transitions, and the wavekind objects contain strings for the names of their relevant resources. For the case of the RZO waveform, the match of the {0,2} case completes with the association of the time value 200ns with the string "T1", and 400ns with string "T2". These pairings are stored in a table corresponding to the STIL character '1' as defined in the WaveformTable.

The WaveformTable-based analysis method takes advantage of economies available when the source format is cycle-based. The information regarding the formats that will be used for the entire process is available up-front and can be processed, analyzed and optimized before pattern conversion begins. Once this has been completed, the processing of the vectors becomes very simple. The STIL pattern character is accessed (S21 in Figure 12), the previous signal value (stored at the end of the previous cycle) is recalled (S22 in Figure 12) and these data are used to access the information stored for the {starting value, STIL pattern character} pair, i.e. the wavekind and parameter information determined from the matching process (S23 and S24 in Figure 12). Then, the ending value is stored for beginning of the next cycle (S25 in Figure 12).

This is markedly different from the situation where the source format is event-based, for example, VCD (Value Change Dump) data by Verilog. In this case, the wave format information is usually not available up front. Matching of source data to target wave shapes must be done as the vectors are processed. This can lead to poor choices that are avoided using proper table analysis techniques.

Assuming the following two cycles in a waveform: If the template matching were done on a cycle-by-cycle basis (as would be the normal case without a table-based analysis), the first cycle would likely end up being mapped to NRZ; this is

the simplest waveform that can satisfy the constraints. The second cycle is clearly RZO. If it were known about the second cycle when processing the first, it would have noted that the first cycle can also be mapped to RZO, and would have prevented an on-the-fly wavekind switch. A table-based analysis approach provides just this capability.

Output Signal Conversion

The preceding description has focused on the mapping of input STIL characters to TDL equivalents. The "input" here means a test pattern supplied (drive) to a pin of device under test (DUT). As previously noted, the test vectors include the test patterns (input) and strobes (output or compare). A strobe is a timing pulse either with edge (no pulse width) or window (predetermined pulse width) to sample the device output signal. Here, the output mapping strategies are described which deals with conversion of strobe signals (compare) in the STIL format to the TDL format. The behaviors specified in the STIL file can be directly translated to pattern characters. For example:

LHZX {'0ns' Z; '0ns' X; '260ns' L/H/T/X;}

can be mapped directly to edge strobes for the appropriate value, if all are applicable. In other words, the template matching is unnecessary in this process. Similarly:

LHZX {'0ns' Z; '0ns' X; '260ns' l/h/t/x; '500ns' x}

can be mapped to the appropriate window strobe.

There are several points worth noting about the output conversion. The first is that some ATE may not allow switching of strobe type on-the-fly. Thus, edge and window strobes may not be mixed. A solution in this case is to make all strobes into window strobes, with edge strobes made into window strobes as narrow as allowed by the system. The second point is that STIL pattern characters that request multiple strobes within a cycle may or may not be compatible with the target ATE system. The capabilities of the ATE

system family is built into the conversion tool with a resource file read at run time to indicate which subset of these features is actually present on a given target ATE system. This would include behaviors such as transition and double strobe modes.

Bidirectional Signal Conversion

Bidirectional signal conversion provides the most complexity in the entire process. Here, the term "bidirectional" means conversion of test language from the STIL formate to the native language format for the test vectors assigned to device pin which functions both input and output. All of the features required for input and output matching noted above are present as well as additional features unique to bidirectional signals. In addition, bidirectional signals often place limitations on tester resources above and beyond those placed by simple input or output signals. As an example, the number of available edges may be reduced due to the need for driver control signals to determine the directional state of the bidirectional signal.

The considerations for driver control are based on the characteristics of the STIL pattern characters and the capabilities of the target ATE system. A standard paradigm is to provide two driver enable modes, one that mimics the NRZ behavior, and one for RZ. In the former case, the cycle becomes "drive" (input) at some point and remains that way through the end of the cycle. For the RZ, the cycle becomes "drive" and then "compare" (output) during the cycle. Note that the cycle being in "compare" mode does not mean that a comparison is actually taking place, just that the pin is to be treated as an output. This distinction becomes important with regard to target ATE capabilities for drive and compare in a cycle.

The driver enable mode is determined from the STIL pattern characteristics by noting that an NRZ driver enable

mode is preferred since it requires fewer tester resources. This mode is chosen unless specifically required to use the RZ mode. This only happens when a "drive" region is surrounded by "compare" regions in the same cycle. Again, actual comparisons may not be occurring, but the device pin is acting as an output. The time values for the driver control edges are determined from the transition times for the signal direction.

The inclusion of driver type information makes the "matcharray" process described above quite a bit more complex. The drive portions of the signal are matched against the available waveforms while the overall character of a cycle, in terms of "drive" and "compare" portions are compared against the capabilities of the target ATE system.

STIL contains the concept of the "DrivePrior" event. This is meant to contain the most-recently-used drive value on a system. For input signals, the prior drive value is always the last state of the signal, so this need not be considered. Output signals have no drive states, so this is irrelevant. Thus, the DrivePrior is only relevant for bidirectional signals where it represents the last drive state attained by the signal, regardless of any intervening strobe activity. On some ATE systems, the presence of a strobe character affects the state of the driver in the system. For example, if the last "drive" state is "D", and this is followed by a strobe for "H", the driver can be set to a "U" state for the next drive cycle. The "DrivePrior" is intended to handle this case. With regard to the conversion process, this prior state must be kept as well as the actual current state of the driver. The result of this is that the "matcharray" shown above contains four columns instead of two, corresponding to all possible pairs {previous value, DrivePrior}, i.e., the column heads (Figures 10A and 10B) are {0,0}, {0,1}, {1,0}, {1,1}.

The data used in these entries must be used properly in order for the concept of the "DrivePrior" to work properly. The desired initial state, as given by the "DrivePrior" value, must be reconciled with the actual state of the driver, previous value. Using the example above, the driver is in the "U" state, and the STIL pattern character is requesting the device to be in the "D" state based on the inclusion of the "DrivePrior" or "P" event in the description. The waveform chosen to match this pattern character must be able to reconcile these values, driving the pin from the "U" state to the "D" state by the time required. This can result in an additional edge not readily apparent from the STIL pattern character description.

Special Features

In the following, the test language conversion is further discussed as to the processing of special features provided in the target ATE system. These features represent those found on the assignee's ATE system, Advantest Model T6600 IC tester family, but they are quite general and may be found, in some form, on a variety of test systems. Thus, brief discussion is made as to the algorithms used to map the STIL information to these features.

Multi-Clock Signals

The multi-clock (MCLK) signal type is commonly used for providing more pulses per cycle than the ATE system can theoretically provide. This is done for repetitive waveforms by essentially breaking the tester cycle into a series of subcycles (internally) and providing multiple copies of a basic waveform, one per subcycle. The result is the appearance of a greater number of edges than possible in the rated test cycle.

The key to the use of the MCLK paradigm is that the waveform must contain a basic repeatable unit within the cycle. During the template matching algorithm described

above, the discovery that a STIL pattern character contains too many edges leads to an attempt to match the character using an MCLK format. For this to work, the number of edges must be even (the MCLK format is pulse-based). The constraints that must be satisfied for a single-pulse repeatable waveform are derived with regard to an upper waveform of Figure 13.

In order to have a repeatable unit, it is required that the widths of all pulses be the same. Furthermore, the spacing between the pulses must be equal to the sum of the space (A) before the first pulse plus the space (B) at the end of the last pulse, as shown. This means that it is necessary to have a basic repeatable unit that looks like RZ with T1=A, T2= A+PW, and a subcycle length of A+PW+B. If any of these conditions are not met it is unable to provide a single pulse basic repeatable unit.

It is possible to create a double-pulse basic repeatable unit. The requirements are derived based on the lower waveform of Figure 13. In this case there are two pulses per repeatable block, and it requires that the corresponding pulses be the same width, in this case the pulses labeled PW1 and PW2. The sum of the begin and end space must match the interval between the repeatable unit repetitions, as above (repeatable unit space = A+B). Also, the spacing (C) between the pulses must be the same for all repeatable units. Thus, in this case, a subcycle length is A+PW1+C+PW2+B.

<u>Pin Multiplexing</u>

In the pin multiplexing (PMUX), two tester channels are combined to drive a single pin. This allows the resources for both tester channels to be used for the same signal, enabling drive (input) and compare (output) in the same cycle on ATE systems where this might not otherwise be available.

While the use of PMUX can make the ATE programming task easier, it adds complication to the conversion process by

providing an additional degree of freedom. While this might seem desirable in that it provides flexibility, it makes the search of the wavekind space a little harder because requirements are not precisely pinned down. In the following, discussion will be made as to the implications of PMUX on the various pin types.

(1) Input PMUX

Input pin mapping is performed by a search of the wavekind space attempting to match the capabilities of the wavekind with the requirements of the STIL pattern character. The details of this approach have been discussed above. When an input signal is used with PMUX, the responsibility for matching the requirements of the STIL pattern character is shared between the two tester channels. The issue presented by the added flexibility is one of determining a "requirement sharing" methodology.

One approach taken is that the simplest sharing algorithm is not to share at all. If possible, one pin does all the work and the other does none. To this end, an attempt is made to map all of the edges specified by the STIL pattern character into the first tester channel. The matching that takes place here is identical to that described above for non-PMUX input signals. The difference occurs if the match attempt fails, i.e. there is no wavekind that can satisfy some subset of a STIL pattern character's requirements. Previously, this situation either resulted in an attempt to use MCLK, or the reporting of an error. With the PMUX case, one edge is shifted from the first channel to the second, and the match is attempted for the first channel and second channel separately. The shifting of edges is continued until a match is possible for both channels for all sets of requirements for the STIL pattern character.

Adding to the complication for this algorithm is the need to maintain continuity between the two component channel

behaviors. A given channel "remembers" its final state from the last cycle. The state of the driver to the device, however, has been set by the signals from the other channel. For example, if channel 1 drives the pin to an "H" state, it will have "H" as its previous state. Now, suppose channel 2 drives the pin to a "L" state. When channel 1 drives again, it will "remember" being in the "H" state, but the driver will actually be in the "L" state because of the action of channel 2. Coordination between the two channels comprising the signal is required to prevent this situation.

As an example, consider the following signal of the uppermost waveform in Figure 14, where T_0 denotes a cycle boundary. This signal is very simple and appears to have a NRZ character for the first signal. In fact, the second signal apparently needs to do nothing except stay low. Consider the first pass solution to this problem, shown in Figure 14.

The signal proposed for Channel (Pin) 1 provides the rising edge at T_a, as desired (second waveform from the top in Figure 14). Since there are no edges in the second portion of the signal, Pin 2 need have no edges (third waveform from the top in Figure 14). In the second cycle, Pin 1 remains high as there are no edges. In the second part of the cycle, there is a low-going edge. Since Pin 2 is already in the "L" state, no edge will be generated.

It is clear from this example that the simple approach will not work in this case. The signal will look exactly like the waveform on Pin 1 as Pin 2 has no edges. This has happened because the continuity across channels has not been maintained. Pin 2 needs to be aware that Pin 1 has ended in an "H" state so that subsequent edges on the system, in this case the transition to "L" at T_a in the second cycle, will be handled properly. This can be accomplished by replacing the waveform for Pin 2 above with that shown in the bottom of

Figure 15.

Here the waveform for Pin 2 is brought to the "H" state at the split time. This will have no effect on the composite signal as the driver is already in the "H" state due to the action of Pin 1. It does, however, condition Pin 2 to be in a consistent state with Pin 1. When the low-going edge happens in the second cycle during the Pin 2 portion this will result in a low-going transition on Pin 2 at T_b as desired. This demonstrates the channel consistency problem that must be handled for PMUX signals.

(2) Output PMUX

As with normal pattern matching, the conversion of output signals in the presence of PMUX is simpler than input conversion. The cycle is split into two portions, and strobe edges are assigned to the two channels (pins) based on their position relative to the split time. For simplicity, the split time is chosen as the middle of the period. Strobe edges occurring before this time are assigned to the first channel, and those occurring after are assigned to the second. One notable exception to this rule is window strobes. These may not be split across the boundary between channels. Consequently, the split time is chosen so that the entire window strobe falls within one of the channels.

(3) Bidirectional PMUX

The use of PMUX with bidirectional signals allows features that might not otherwise be available. This is a primary use for the PMUX construct, rather than for pure input or output signals. Probably the single most important use of the PMUX is to allow drive and compare within a cycle, if the target ATE system does not allow this. For cases where a STIL pattern character for a bidirectional signal specifies pure drive or compare behavior for a cycle, the processing is virtually identical to that described with respect to the input and output pin multiplexing above. When

the behavior is mixed, the split time for dividing the cycle between the channels is based on the time of the direction switch. This leads to a very natural division of the responsibilities of the two channels. The concept of continuity across channels that we discussed for input signals applies here as well, but must also take into account the effects of intervening strobes.

As has been described above, according to the present invention, the test vectors in an original test language are converted to a target test language with high efficiency and high accuracy.

Although only a preferred embodiment is specifically illustrated and described herein, it will be appreciated that many modifications and variations of the present invention are possible in light of the above teachings and within the purview of the appended claims without departing the spirit and intended scope of the invention.

WHAT IS CLAIMED IS:

1. A method of converting test vectors in an original cycle based test language into a target cycle based test language, comprising the following steps of:

reading available waveforms defined in the target test language and forming a set of templates depicting the waveforms where each template corresponds to a waveform of the target test language and includes data showing at least a starting value of a segment of waveform and a number of subsequent edges in the waveform;

reading the test vectors of the original test language and decomposing a waveform in the test vectors in the original test language into a set of constituent events where each event includes data showing at least a starting value and a number of subsequent edges of the waveform;

comparing the template derived from the waveform in the target test language and the set of events derived from the original test language;

storing the waveform data in the target test language when a match is detected in the comparison step and retrieving corresponding parameters of the waveform in the test vectors of original test language and storing the parameters in combination with the matched waveform data;

repeating the above steps for all of the test waveforms in the original test language, thereby forming representation of the waveforms in the target test language.

2. A method of converting test vectors as defined in Claim 1, wherein the step of applying the comparison algorithm at different levels of abstraction, in the order of a signal level, a wavekind level where the signal is

configured by a plurality of wavekinds, and a character level where the wavekind is configured by a plurality of characters.

3. A method of converting test vectors as defined in Claim 1, wherein the set of events is stored in a table format having columns assigned to the data showing the number of subsequent edges and the starting value.

4. A method of converting test vectors as defined in Claim 3, wherein the table storing the set of events is optimized by studying the starting value of a particular event based on an ending state produced by the previous event, thereby simplifying the data in the table.

5. A method of converting test vectors as defined in Claim 1, wherein the test vectors includes drive signals to be supplied to a device under test (DUT) as an input and strobe signals to sample an output of DUT for evaluation, wherein the drive signals in the original test language are converted to the target test language by comparing the template and the set of events for detecting the match while the strobe signals in the original test language are directly translated to the target test language.

6. A method of converting test vectors as defined in Claim 1, wherein the waveforms in the original test language are assigned where required by resource limitations to a plurality of subcycles of the target test language where the plurality of subcycles are created by multiplexing a test cycle clock in a test system which is operated by the target test language.

7. A method of converting test vectors as defined in Claim 1, wherein the waveforms in the original test language are assigned to a plurality of test channels of the target test language where the plurality of test channels are multiplexed to be connected to a single pin of DUT in a manner configured by a test system which is operated by the

target test language.

8. A method of converting test vectors in a STIL (Standard Test Interface Language) into a target cycle based test language, comprising the following steps of:

reading available waveforms defined in the target test language and forming a set of templates depicting the waveforms where each template corresponds to a waveform of the target test language and includes data showing at least a starting value of a segment of waveform and a number of subsequent edges in the waveform;

reading the test waveforms of the STIL format and decomposing a waveform in the test vectors in the STIL format into a set of constituent events where each event includes data showing at least a starting value and a number of subsequent edges of the waveform;

comparing the template derived from the waveform in the target test language and the set of events derived from the waveform in STIL;

storing the waveform data in the target test language when a match is detected in the comparison step and retrieving corresponding parameters of the waveform in the test vectors of STIL and storing the parameters in combination with the matched waveform data;

repeating the above steps for all of the test vectors in STIL, thereby forming a test vector file of the target test language.

TEST LANGUAGE CONVERSION METHOD

Abstract of the Disclosure

A method of converting test vectors in an original cycle based test language into a target cycle based test language. The method includes the steps of forming a set of templates depicting waveforms defined in the target test language; decomposing a waveform in the original test language into a set of constituent events where each event includes data showing at least a starting value and a number of subsequent edges of the waveform; comparing the template and the set of events; storing the waveform data in the target test language when a match is detected and retrieving corresponding parameters of the waveform in the original test language; and repeating the above steps for all of the test vectors in the original test language, thereby forming a test vector file of the target test language.

SPC-A214.103
032202